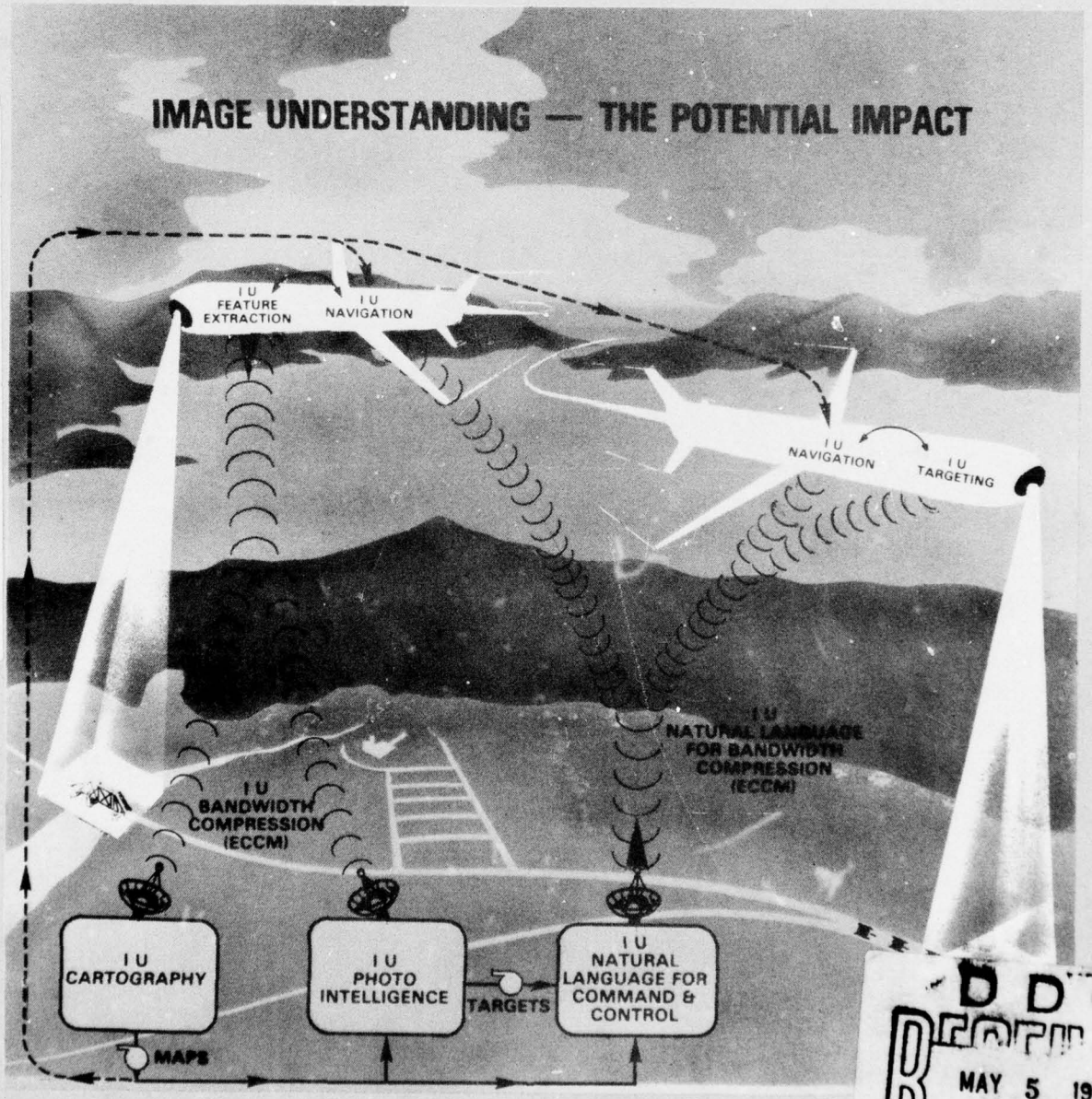
PROCEEDINGS:

IMAGE UNDERSTANDING WORKSHOP

MAY 1978

Sponsored by:
Information Processing Techniques Office
Defense Advanced Research Projects Agency



Science Applications, Inc.

IMAGE UNDERSTANDING

Proceedings of a Workshop held at Cambridge, Massachusetts, May 3-4, 1978

Sponsored by the
Defense Advanced Research Projects Agency

Science Applications, Inc.
Report Number SAI-79-749-WA

Lee S. Baumann
Workshop Organizer and
Proceedings Editor

This report was supported by the Defense Advanced Research Projects Agency under DARPA Order No. 3456, Contract No. MDA903-78-C-0095 monitored by the Defense Supply Service, Washington, D.C.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SAI-79-749--WA	2. GOVT ACCESSION NO. ADA 052902	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Proceedings: Image Understanding Workshop, May 1978		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical October 1977-April 1978 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lee S. Baumann (Ed.)		8. CONTRACT OR GRANT NUMBER(s) MDA903-78-C-0095
9. PERFORMING ORGANIZATION NAME AND ADDRESS Science Applications, Inc. 1911 N. Fort Myer Drive, Suite 1200 Arlington, Virginia 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order No. 3456
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE May 1978 13. NUMBER OF PAGES 158
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Copies of the April 1977 Proceedings are available from DDC under accession No. ADA 052900. Copies of the October 1977 Proceedings are available from DDC under accession No. ADA 052901.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Image Processing, Image Understanding, Scene Analysis, Edge Detection, Image Segmentation, CCD Arrays, CCD Processors.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document contains the technical papers and outlines of semi-annual progress reports presented by the research activities in Image Understanding sponsored by the Information Processing Techniques Office; Defense Advanced Research Projects Agency. The papers were presented at a workshop conducted on 3-4 May 1978 in Cambridge, Massachusetts.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407 154

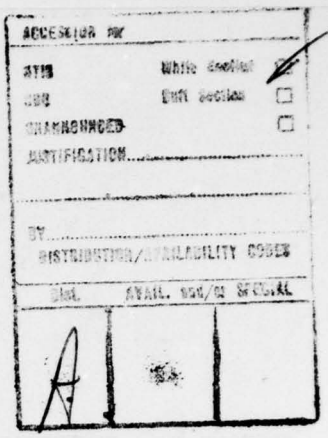


TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

*Paper not presented

FORWARD

With the convening of the Seventh Image Understanding Workshop in Cambridge, Massachusetts on 3-4 May 1978, the planned five year Defense Research Projects Agency program reached its temporal mid-point. It is interesting to note that this program was initiated by the calling of a preliminary workshop in March 1975 which attempted to set out goals for both the research effort as well as the areas in which the results might make significant impacts in the military user community. At that initial workshop, Dr. J. C. R. Licklider, then the Director of the Information Processing Techniques Office which sponsors the program, made this observation:

> "The objective (of the Image Understanding Program) will be to develop the technology which can be exploited by the DoD components to solve their specific problems. Thus, the activities that will be supported in the program will not be the engineering of specific solutions to specific problems. The philosophy in the program will be to develop generalized technology by driving the research in particular directions. However, at the end of the five year period the technology developed must be in a state in which it can be utilized by the DoD components to solve their specific problems without requiring a significant research effort to figure out how to apply the technology to the specific problems. For this reason, the program must result in a demonstration at the end of the five year period that an important DoD problem has been solved."

Also at the initial meeting, LTC Carlstrom, the Program Manager for the Image Understanding Program, presented a list of potential problem areas of interest to Image Understanding as follows:

1. Photo Intelligence
2. Geophysical (ERTS, LANDSAT)
3. Cartography
4. Meteorology
5. Remotely Piloted Vehicles (Robotics, Guidance)
6. Surveillance

In the two and a half years of its existence, the DAPRA Image Understanding Program has attempted to follow the philosophy enumerated at its founding as cited above. The original meeting was attended by 31 representatives, while workshops during the first year of research attracted 50 attendees. At the end of the second year the size of the Image Understanding Workshops had grown to 72 interested personnel with many more receiving copies of the workshop proceedings. It is hoped that this increase in interest is a reflection of the emphasis that the program manager and the research personnel have placed on demonstrable and real world results. The close interaction of the user community by attendence and participation at these workshops is much appreciated by all concered with the program.

This document contains technical reports presented by those research organizations active in the Image Understanding Program. Also, outlines of the semi-annual progress reports as presented by the Principal Investigators are included for reference. The University and Industrial companies currently working on the DARPA program are:

- University of Southern California
- University of Maryland
- Purdue University
- Carnegie-Mellon University
- Massachusetts Institute of Technology
- Stanford University
- University of Rochester
- SRI International
- Hughes Research Laboratories
- Westinghouse, Incorporated
- Honeywell, Incorporated
- Control Data Corporation
- Lockheed Missiles and Space Company

The seventh workshop was hosted by Dr. Patrick H. Winston, Director of the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. The meetings were conducted at the Howard Johnson's Motor Lodge, Cambridge, Massachusetts, on 3-4 May 1978. Representatives of various Army, Navy, Air Force, and DoD and other Government Agencies as well as members of the research organizations concerned were in attendence. Thus the two primary objectives of the workshop - the cross-fertilization of research results among the various investigative groups and an exchange of ideas between the users and the research personnel - were accomplished. It is particularly gratifying to note that several demonstrations of achieved research results were presented at this workshop in addition to the descriptive technical papers.

The cover design of this volume attempts to carry forward the hierarchical processing theme and the multiple technologies theme of the past two proceedings by indicating a possible direction for the final utilization of the products of this research program, i.e., actual technology transfer from the laboratory to the field. Although DARPA does not concern itself with the fielding of systems - it is vitally concerned that its research efforts be ready for use by service or DoD agencies. The artwork for the cover was created by David E. Badura and Thomas G. Dickerson of Science Applications, Inc. from ideas supplied by LTC Carlstrom.

The Conference Organizer wishes to thank the moderators of the technical sessions for keeping the program on schedule and Dr. Winston of MIT for hosting the workshop and arranging tours and demonstrations of the MIT-AI Laboratory. Ms. Suzin Jabari of the MIT staff was instrumental in making the arrangements for the workshop in the Boston area. Mrs. Kristin G. Johncox of Science Applications, Inc. provided typing support for mailings and the collection and arrangement of the conference proceedings.

Lee S. Baumann
Science Applications, Inc.
Workshop Organizer

Reference:

1. Minutes of the 6-7 March 1975 Meeting of the ARPA Image Understanding Workshop, Page 3, Page 10.

AUTHOR INDEX

SESSION I

HARDWARE

HARDWARE IMPLEMENTATION OF A SMART SENSOR: A REVIEW

Thomas J. Willett
Glenn E. Tisdale

Westinghouse Systems Development Division, Baltimore, Maryland 21203

ABSTRACT

This paper summarizes the results of a 21-month program to investigate the design and implementation of automatic target cueing logic. The work was performed for the University of Maryland Computer Science Center, under DARPA sponsorship, and monitored by the Army's Night Vision Laboratory.

During the conception and test by Maryland of the cueing algorithms, Westinghouse carried out an investigation of techniques for their implementation, with particular emphasis on charge transfer devices. When processing functions were specified, a detailed analysis was then carried out so as to provide them in CCD's. This process continued throughout the first year of the program.

During the final nine months, a specific circuit was chosen for the fabrication of a demonstration unit. A sorter function was selected because of its occurrence in several cueing operations. Chips were fabricated and tested at the Westinghouse Advanced Technology Laboratories, and a demonstration unit was assembled and shown at the Image Understanding Workshop in October, 1977. The unit rearranges a random series of pulses in ascending order by magnitude.

An estimate was also made of the area in monolithic silicon required to implement the cuer function in CCD's. The algorithm presently proposed by Maryland would require an area of 11-1/4 inches by 7-1/2 inches. If 3-inch by 3-inch modules were employed with 1/2-inch centers, an equivalent volume would be 3 inches by 3 inches by 6 inches.

INTRODUCTION

Although the sensors used in reconnaissance and target acquisition continue to improve in resolution,speed, and dynamic range, the location of targets still depends on the ability of a human operator to search images in real time. The concept of the "Smart Sensor" assumes that much, if not all of the human effort in target acquisition can ultimately be performed better by automatic recognition logic. An initial step in the development of the Smart Sensor would provide machine assistance to the human in evaluating his displays, by providing audible and visual cues regarding target location and identity. Development and implementation of automatic target cueing algorithms was the subject of a 21-month program performed with the University of Maryland Computer Science Center. The program contained three phases, as follows:

Phase I: Task and Technology Review (3 months)

Phase II: Algorithm Selection and Test (9 months)

Phase III: Hardware Development (9 months)

All phases of the program were completed during the prescribed period, including the construction and demonstration of an important recognition function using charge-coupled devices.

The success of the program is due, in large part, to the close coordination between members of the government-university-industry team. The team was assembled in 1976 by Lt. Col. David Carlstrom of DARPA. The principle team members from the government were Mr. John Dehne and Dr. George Jones of NVL. For the University of Maryland, principle members were Profs. Azriel Rosenfeld and David Milgram. The Westinghouse team included Dr. Glenn Tisdale, Program Manager, Mr. Thomas Willett, project engineer, Dr. Nathan Bluzer, and Dr. Gerald Borsuk.

The design of an automatic target cueing system must begin with a statement of system design goals. Next, the algorithms and data flow can be established. Finally, hardware fabrication can be considered. This paper will summarize the effort in each area.

SYSTEM DESIGN GOALS

As shown by Fig. 1, automatic cueing is achieved by an image processor, which serves as an information filter on the image, alerting the human to the presence of potential targets, possibly by audible signals initially, and then by providing visual cues or overlays on his display. Automatic cueing can be carried out either in airborne or ground locations. In the airborne situation, the operator views a CRT-type display for acquisition of targets on a real-time basis. His determination may result in action in a matter of seconds, either offensive or defensive. On the ground, interpretation may be required in real-time, or on a more relaxed basis. In the proposed

operation of remotely piloted vehicles (RPV's), for example, a video link may be used to obtain a CRT presentation at a ground station of the output of a sensor located on the vehicle. The problems for the operator are somewhat similar to the airborne situation; however, his appraisal of the sensor image is entirely limited to the CRT output. He can't look at the target area directly. On the other hand, he is not distracted by problems such as vehicle operation and personal security.

Key considerations in the design of an automatic target cueing system are its performance, physical characteristics, and allowable cost. A quantitative determination of design parameters will depend on the manner in which the mission is implemented. Such implementation will be discussed first.

As explained above, the target cueing function might be performed aboard a vehicle, or at a ground station if imagery is relayed for analysis. In either case, the performance goals will tend to be comparable. As regards physical characteristics and cost, however, the vehicle location will demand much tighter restrictions. Our discussion will proceed on the basis of the vehicular application. Both helicopters and high-speed aircraft are airborne candidates. The RPV image, on the other hand, will be analyzed at a ground station; therefore, the physical limitations within the RPV are not a problem. As the state-of-the-art in automatic target recognition develops, and high levels of performance are attained, it is anticipated that the human observer will eventually be eliminated in some applications. For example, recognition equipment might be placed aboard a missile for unaided terminal guidance. The requirement for high performance, small size and weight, low power consumption, and low cost will all apply in this case.

Performance Goals

Key performance parameters are the detection and recognition rates for targets of interest, the false alarm rate, and the speed of operation of the cueing system. Detection occurs when a target of any kind is indicated by the cueing system, while recognition occurs when the correct target class is selected from among several possible classes. Detection and recognition performance is expressed as a percentage of the targets which are actually available. A false alarm occurs when a target is indicated even though none is present. The false alarm rate is expressed on the basis of a unit of elapsed time or area of coverage. The required speed of operation is determined by the time available to the operator to make decisions, the search area to be covered by the sensor, and the sensor frame rate. It depends heavily on human factors considerations, such as decision times and reaction times, and the choice of prioritization ground-rules.

A detailed examination of the trade-offs between the required cuer processing rate and the allowable false alarm rates was presented recently by Dehne et. al. of NVL[1]. For a set of mission parameters which relate primarily to the helicopter scenario, it was found for processing rates between 3 and 10 frames per second, false alarms could be accommodated increasing from 0.5 to 1.8 per frame. These results assumed that 20 seconds were available to cover a large search window, resulting in about 0.7 seconds to handle each frame. This figure includes the frame processing time, the time to slew the sensor, the operation decision time per false alarm (0.2 seconds) and his reaction time to advance to the next frame (0.2 seconds). The report considers sequential as well as combined processing and slew, with the former preferred.

A separate consideration with the above processing rates is that the cueing symbols superimposed on the display must appear in the correct location even after the processing delay. With a frame rate of 30 per second, the delay covers 3 to 10 frames (0.1 to 0.3 seconds). Misregistration of target and symbol could be caused by target motion relative to the terrain, or the changes in the field of view due to aircraft motion. Broadside motion of the target is generally the worst case. Suppose the cueing window subtends twice the extent of the target on the display in both the horizontal and vertical dimensions, and is located at the point of the target center in the processed frame. It can move by half its dimension in any direction and still be contained within the cueing window. For a vehicle 20 feet in length which subtends 20 pixels in the display, motion of 10 feet must be accommodated over a worst case period of 0.3 seconds, with a corresponding allowable broadside speed of 30 feet per second (43 mph). This result is independent of range if the window is porportional to target size.

The report also considers the use of a wide sensor field of view for cueing, followed by operator confirmation with the narrow field of view. It is assumed in this case, that the capability of the cuer for recognition exceeds that of the operator by an amount sufficient to compensate for the increased field of view. Under these conditions, one false alarm per frame could be accommodated with a processing rate of 0.54 frames per second (about a 10:1 reduction over the previous case). However, at the present state of the art, this improved cuer performance relative to the operator has not been demonstrated. In that regard, it is noted that because of the eye integration time of 0.2 seconds, the operator gains a signal-to-noise improvement over the cuer of, perhaps, 2.5.

A final approach considered a sensor with an expanded, high resolution scan area equivalent to the target search window. Due to display limitations, the operator sees either a low

1.Dehne, J.S., Van Atta, P. and Raimondi, P, Specifying Image Processing Systems for Thermal Imagers, paper presented to the Seventh Annual Symposium of the EIA-AIPR Committee, College Park, Md. 24-24 May 1977.

resolution version of the entire window, or a small segment containing potential targets as selected by the cuer. The cuer processing rate is not greatly affected by this mechanization.

For the assumed frame size of 375 by 500 pixels, the processing rates of 3 to 10 frames per second correspond to data handling rates of 0.6 to 1.9 megapixels per second.

The foregoing discussion was addressed to the helicopter scenario. For the high-speed aircraft, the available search time will tend to be lower, but the required search window will probably also be reduced. The reduced search window can be achieved by reliance on navigation aids for the acquisition of predesignated targets. At the high speeds and possible low altitudes, it appears that the single-seat operator, because of the burden of aircraft navigation, will be assisted significantly by the cueing system. Frame rates which are comparable to his reaction time, or somewhat lower, in the vicinity of two per second should be tolerable from the point of view of overall processing time. However, from the point of view of increased detection and recognition rates, as well as reduced false alarm rates, it appears useful to consider the integration of results from successive frames. The assignment of a priority weighting to target cues will improve the probability that important targets will be considered when a number of opportunities occur.

Physical Characteristics

The significant physical characteristics of the cueing system for aircraft or missile use include size, weight and power consumption.

The increased availability of general-purpose MSI circuits has made it possible to offer existing cuer algorithms, using conventional packaging techniques, in packages which should be acceptable for aircraft use. A total system, excluding displays, might be expected to occupy a volume of 0.5 to 1.0 cubic feet, and to weigh 10 to 30 lbs., including power supply. Power in the neighborhood of 200 to 300 watts will be required.

For missile applications, conventional packaging can be improved upon by use of flat packs, or bare chip packaging, and by the introduction of some specially designed chips.

One thrust of the present Westinghouse program, however, has been to determine the necessary area of silicon substrate required to provide cuer functions. As will be described later, the fabrication of special CCD LSI circuits appears feasible, and would reduce the cuer to an overall chip area measured in square inches. On this basis, introduction of cueing functions into an artillery shell, for "fire-and-target" performance, becomes feasible.

Allowable Cost

Since the cueing system is a digital processor, its production cost using conventional packaging techniques can be estimated reasonably well. For helicopter or aircraft use, a figure of $20K to $50K per unit is suggested. Reductions in size by increased use of LSI will tend to reduce the recurring cost for each unit, but at the expense of a significant nonrecurring cost for initial development.

Implementation of a complete cueing system, using CCD circuits on a small number of silicon chips, takes this sequence one step further. The final unit recurring cost, in production, might range from $1K to $5K, including test, but the development program would be a multimillion dollar affair. Before such an investment could be considered, a number of hurdles would have to be overcome. First, the satisfactory operation of the system would have to be established. Next, an attempt should be made to compare the performance of competitive approaches, since only one design can probably be initiated. Finally, a variety of applications should be considered, so that the development cost can be divided as much as possible.

A practical approach to this dilemma, which has been initiated in the present program, consists of the selection and implementation of key circuit functions in CCD form. These circuits can hopefully be used in hybrid arrangements to reduce the size, weight, and power of early cuer designs. With the growing availability of new chips, these values will continually decrease. At the same time, the solution of a variety of application problems will be possible from the library of available chip designs.

ALGORITHM IMPLEMENTATION

We now turn to a preferred set of algorithms developed by the University of Maryland which comprise the first portion of a cueing system. A system flow chart is shown in Fig. 2. In general, the Median Filter acts to suppress noise. The Gradient Operator extracts edges which are then thinned by the Non-Maximum Suppression Algorithm. At the same time a set of gray levels is determined and the filtered image is thresholded at each gray level. A Connected Components Algorithm partitions each of the thresholded images into potential object regions. The Super Slice Algorithm correlates perimeter points formed independently by the Non-Maximum Suppression and Connected Components Algorithms and a score is obtained for each gray scale threshold. These scores and several other algorithms form a set of Classification Logic.

The Median Filter is the first algorithm performed and acts to extract the median gray level from a 5 x 5 array of pixels and to place that median value in the center of the 5 x 5. The Filter quantizes each of the 25 analog signals into a number of discrete units and then sorts

the quantized signals by arranging them in decending order by magnitude. The Filter acts as a moving 5 x 5 window across the image in that having obtained a median value, the first column is dropped and a sixth column is added.

The silicon substrate forming the Median Filter Quantizer is shown in Fig. 3. Din is the diffusion diode through which charge is injected into the chip and into the holding well, HW. DC blocks the charge from leaving via Din. An amount of charge, Q, proportional to S, the signal voltage, is removed from HW via the transfer gate, TG, and placed in the signal well, SW. Via the blocking gate, BG, and the thimble well, TW, a descrete amount of charge, q, is removed and placed in well g1. Another quantum q is removed from SW and placed in g1 while the first charge has been shifted to g2. This process is repeated until SW is empty and all the charge has been placed in a number of discrete wells g1, g2, ... gn.

Recall that the contents of wells g1, g2, ... gn of the Quantizer each contain, at most, an amount of charge q. The content of each well is parallel shifted into its corresponding channel of the Sorter (Fig. 4) so that if there were q charge in g3, there is now q charge in I3 and so on. Forming traps with wells Pg1, Pg2, and Pg3, the charge in each channel is shifted into the large holding well (LHW). The large holding well is partitioned into N channels also which are pulsed simultaneously in charge removal. This means that the numbers are removed from the large holding well in a descending order by magnitude.

The Gradient Operator Algorithm computes edges based on an image of median values; it computes an operator, OP = max {|A-B|, {|C-D|} based on four overlapping regions A, B, C, D each of which consists of 4 x 4 pixels and are arranged in the shape of a cross. The quantities A, B, C, D in the expression represent the sum of all sixteen pixels within each region. The operator OP also works as a moving window and the computational result is placed in the center pixel location. The key arithmetic operation in GRAD OP is the formation of the difference

$$A\text{-}B = \begin{cases} 0 & \text{if } |A| < |B| \\ A-B & \text{if } |A| > |B| \end{cases}$$

which is realized on a silicon substrate with the configuration shown in Fig. 5. Din is a diffusion diode through which charge is injected into the chip; A and B are gates whose potentials are controlled by voltages representing the sums A and B respectively. These gates will form a trap to retain some of the injected charge. The trapped charge is equal to A-B and is removed by the transfer gate. The algorithm

$$B\text{-}A = \begin{cases} 0 & \text{if } |B| < |A| \\ B-A & \text{if } |B| > |A| \end{cases}$$

requires another silicon substrate in which the gate positions of A and B are reversed. The block diagram of the absolute difference operator |A-B| is shown in Fig. 6. A similar block can be formed for |C-D| and then the two connected in a straight forward manner to form GRAD OP = max {|A-B|, |C-D|}.

The Gradient Operator extracts edges in either the horizontal or vertical direction; the Non-Maximum Suppression Algorithm then looks in a direction perpendicular to the edge for a larger gradient value. If a larger edge cannot be found, the edge under consideration is retained; the edge is removed if a larger value is found. Embodiment of the Non-Maximum Suppression Algorithm requires several operations with CCD structures. A key part of NMS is extracting the largest, Xm, gradient value in the neighborhood surrounding the pixel of interest, y . Xm is then compared to the gradient value yg representing the yth pixel. Sorting the X values to obtain Xm can be accomplished by the sorting operation described earlier. Comparing Xm and yg can be done by the subtraction module also described before. Then the block diagram appears in Fig. 7. This time the subtraction module outputs an enable signal to the CCD shift register instead of the actual difference. A blocking gate is used to block (enable) yg from entering the register.

Moving to the right side of the System Flowchart (Fig. 2) we now discuss Threshold Determination. The philosophy here is not to attempt to find a single threshold but rather a set of thresholds which span the range of gray scale. For the NVL FLIR data, fifteen (15) levels represented the gray scale range and selecting every third (3rd) level as a threshold was deemed to give satisfactory target detections by the University of Maryland. Implementation of this type of algorithm requires a sorter which arranges the gray levels in descending order. The first number (the largest) leaving the sorter, and corresponding to the first threshold, sets a counter to 1. The second exiting number is then compared with the first and the counter is updated to 2 if they are different. If it is the same, the counter remains at 1. In general, each number is compared with the previous one to determine if the counter should be updated. When the counter goes to 4, the second threshold has been determined. In this manner every third gray scale is selected and used as a threshold. A block diagram of the implementation is shown in Fig. 8.

The purpose of the Connected Components Algorithm is to segment an image into object regions; these object regions are potential shapes of interest and features are extracted from them for classification purposes. The Operator moves along an image line, with the previous line in memory, determining which pixels are in a specific object region or if a new object region is starting. Each pixel can be examined with respect to its neighbors to the left and above. No diagonal connections are permitted under this convention, and adjacent (horizontal or vertical) pixels must be occupied in order to make a connection. No

skips or gaps are allowed. If we are to extract features from each object region, there must be a means for distinguishing between different object regions. One approach to the problem is to paint (assign an analog voltage level) each object with a different color (analog voltage level) and then have a feature extractor assigned to each color (voltage level). Where an object has several colors, the feature extractors corresponding to those colors cummulate their features, dump them in a scratch feature extractor to combine them, and reassign the results to one of the two feature extractors.

The system block diagram is shown in Fig. 9. The delay line is represented by twenty (20) SI/SO CCD delay lines which are coded to obtain 100 colors (analog voltages) and obviate transfer efficiency problems. There are 20 levels of color comparisons for horizontal and vertical connections in the Coloring Operator shown in Fig.10. The Equivalence Box notes horizontal and connections between different colors, recolors a pixel if necessary, and notes when a color is no longer used thus activating the equivalence statement between two different colors. The switching matrix and latches for the Equivalence Operator are shown in Figures 11a and 11b. The column clock is actually fed to all the Feature Extractors and they indicate when a color is no longer used. The Feature Extractors which accumulate the object features such as area and perimeter as well as the Scratch Feature Extractor form the basis for classification decisions. The Feature Extractors are visualized as a many channeled, large holding well which follows along the line of the histogrammer-sorter. Each channel would correspond to a particular feature and since the features are cummulative, they would simply add in the Scratch Feature Extractor.

DATA FLOW

The Median Filter, Gradient, and Non-Maximum Suppression Operators are calculated for small windows which move over the entire frame. These windows are formed by parallel shifting one line of image from the TDI array into a parallel in, serial out shift register. This register and others then form a serpentine delay through which the pixels are shifted. Non-destructive readouts form the regions comprising the appropriate window. It appears that the computation speed of the Median Filter, Gradient, and Non-Maximum Suppression Operators is conservatively estimated at 50-100 KHZ, hence a parallel organization of the focal plane is necessary for a 1 megapixel/sec. rate. Appropriately, we divide the focal plane into 20 vertical sections each with its own set of Operators (see Fig. 12), to avoid numerical integrity problems. Such a division, however, is undesirable in the Connected Components case because image reconstruction becomes very difficult. Further, the imput data is binary so numerical integrity problems are not present and the CCD implementation of the Connected Components Algorithm can operate at 1 megapixel/sec. Hence, there are five Connected Component Modules corresponding to each of five thresholds.

FOCAL PLANE AREA

Here, we present a preliminary estimate of the focal plane area occupied by the cueing system described. That is, the image has been smoothed (Median Filter), edges obtained (Gradient Operator), edge with reduced (Non-Maximum Suppression), the image has been segmented in object regions (Connected Components), a best threshold selected (Super Slice), and features extracted (area, height, width, perimeter extent, average gray level). The estimate is preliminary in the sense that none of the clocking circuitry has been included in the estimates for the operators. The estimates do not include the Classifiers, although their area contribution will be quite small.

Assuming that the focal plane is divided into 20 columns Fig. 13 shows the number of processors required for a system data rate of 1 megapixel per second. It also shows the geometric area required for each processor and an estimate of the area as defined above. The area is 11 1/4 inches by 7 1/2 inches which is equivalent to a 3 x 3 x 6 inch volume.

CHIP DEVELOPMENT

Figure 14 shows the algorithms developed by the University of Maryland and the functions which are required by each algorithm. A perusal shows that the sorter function occurs in four out of the five algorithms and is the one we selected. Several versions (buried channel and surface channel CCD) of the sorter were put in production runs of the Westinghouse Advanced Technology Laboratory. Figure 15 shows a wafer of the buried channel devices. One portion of the demonstration unit is shown in Figure 16, with the device mounted in place. The ten thumbwheels represent the unsorted numbers which the sorter must rearrange in ascending order. The observer may dial in any arrangement of numbers which he wishes. The outputs and inputs, i.e., the unsorted and sorted arrangements are shown on a two trace oscilloscope. Westinghouse IR&D accounted for 70 cents of every dollar spent on the Smart Sensor Project.

The demonstration unit and a two-trace oscilloscope were exhibited at the DARPA Symposium in September of 1977 at Stanford University in Palo Alto, California. The symposium participants were encouraged to dial in their own set of random numbers on the thumbwheel switches and observe the ordered results. Figure 17 is a typical trace; the random sequence is shown in the left half of the trace and the ordered sequence on the right. The unit was also demonstrated at the Night Vision Laboratory, Ft. Belvoir, Virginia on November 28, 1977.

CONCLUSIONS AND RECOMMENDATIONS

This work has shown that the Smart Sensor can be implemented with CCD technology in a smaller package than implementation with digital techniques. Further, higher level operators (segmentors), normally thought to be only implementable with conventional digital techniques, can be implemented

in CCD.

The total area estimate is 11 1/4 inches by 7 1/2 inches. If 3 inch by 3 inch boards were employed with 1/2 inch centers, the volume dimensions would be 3 inches by 3 inches by 6 inches.

The next step in proving feasibility involves building some of the modules and checking them for numerical integrity, size, speed, and power consumption. These modules, e.g., Median Filter would probably be hybrid packages in the first build, and the clocking circuitry included on the chip. Other items of particular interest are the Connected Components Algorithm with the switching matrix and peripheral control logic and a histogrammer which is derivable from the sorter. Estimates of ultimate size for the monolithic elements would be necessary as well as estimates on the groupings of elements within the monolithic packages.

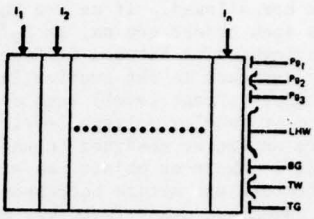


Figure 4: CCD Sorter

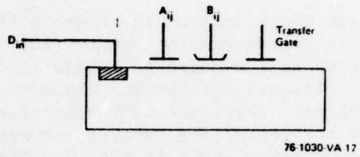


Figure 5: CCD Subtraction Block

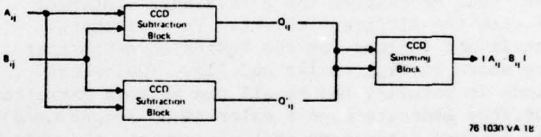


Figure 6: CCD Absolute Value Operator

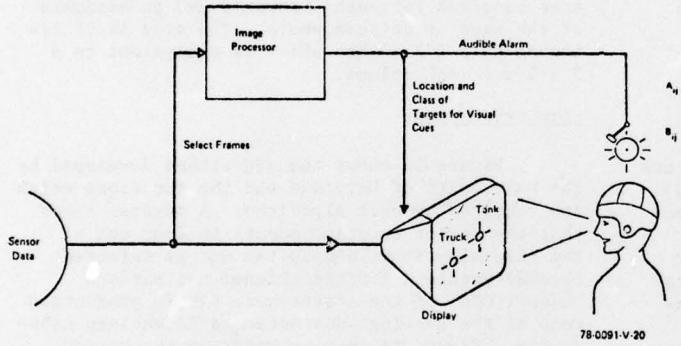


FIGURE 1: The Automatic Cueing Function

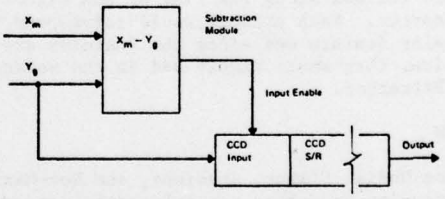


Figure 7: NMS Block Diagram

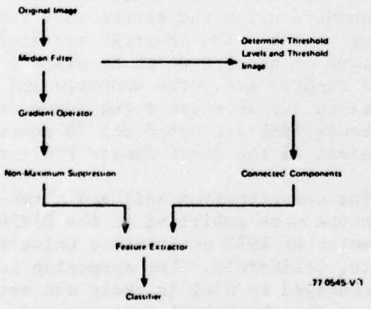


Figure 2: System Flow Chart

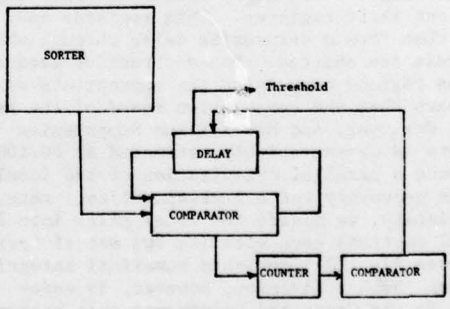


Figure 8: Threshold Determination

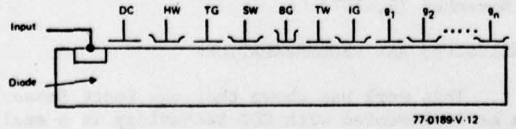


Figure 3: Charge Quantizer

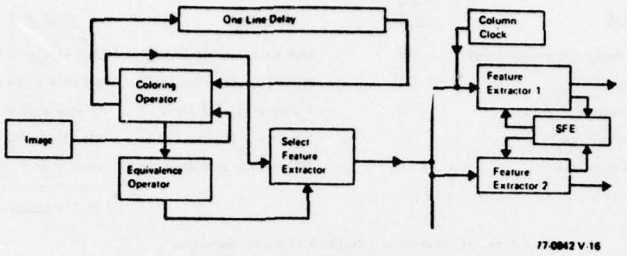


Figure 9: System Block Diagram

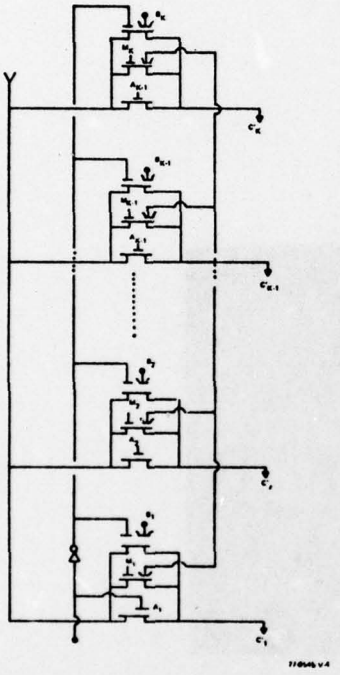


Figure 10: Details of Module C

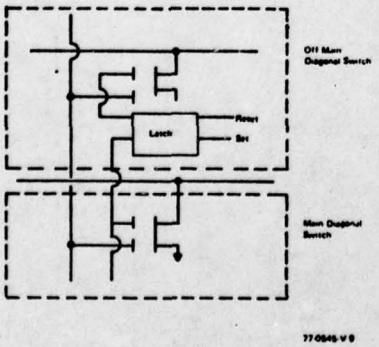


Figure 11b: Switches within the Target Sorter

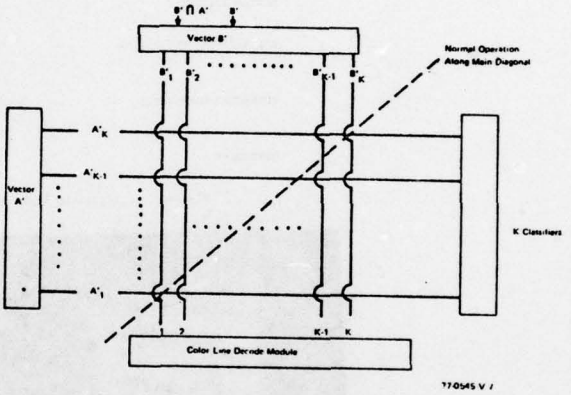


Figure 11a: Target Sorter

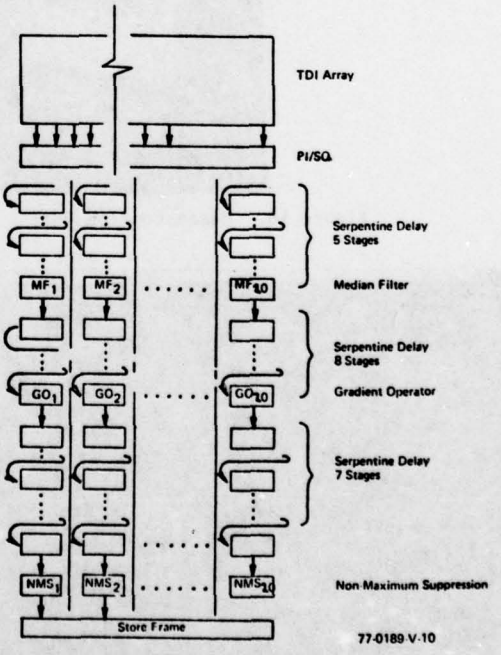


Figure 12: Focal Plane Data Flow

PROCESS	NUMBER REQUIRED	UNIT AREA	TOTAL AREA
Serpentine Delay (36 pixels long)	400	1000 mils x 1 mil for 20	1000 mils x 20 mils
Median Filter	20	100 mils x 128 mils for 1	400 mils x 640 mils
Gradient Operator	20	8 mils x 10 mils for 1	32 mils x 50 mils
Non-Maximum Suppression	20	100 mils x 25 mils for 1	300 mils x 200 mils
Connected Components	5	3750 mils x 3750 mils for 1	11250 mils x 7500 mils
			11 1/4 inches x 7 1/2 inches

Figure 13: Preliminary Estimate of Focal Plane Area

ALGORITHM	FUNCTION
GRADIENT OPERATOR	ABSOLUTE DIFFERENCE COMPARISON
MEDIAN FILTER	QUANTIZER SORTER
NON-MAXIMUM SUPPRESSION	QUANTIZER SORTER ABSOLUTE DIFFERENCE
CONNECTED COMPONENTS	QUANTIZER SORTER COLOR LINE
HISTOGRAM	QUANTIZER SORTER

Figure 14: Algorithm Implementation by CCD Function

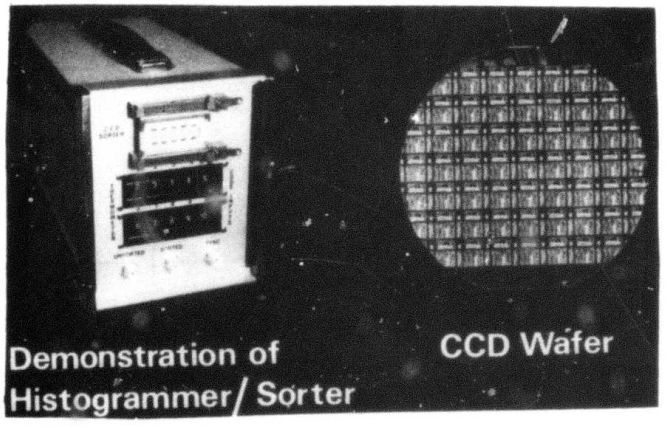


Figure 16. Demonstration Unit

Figure 15. Buried Channel Wafer

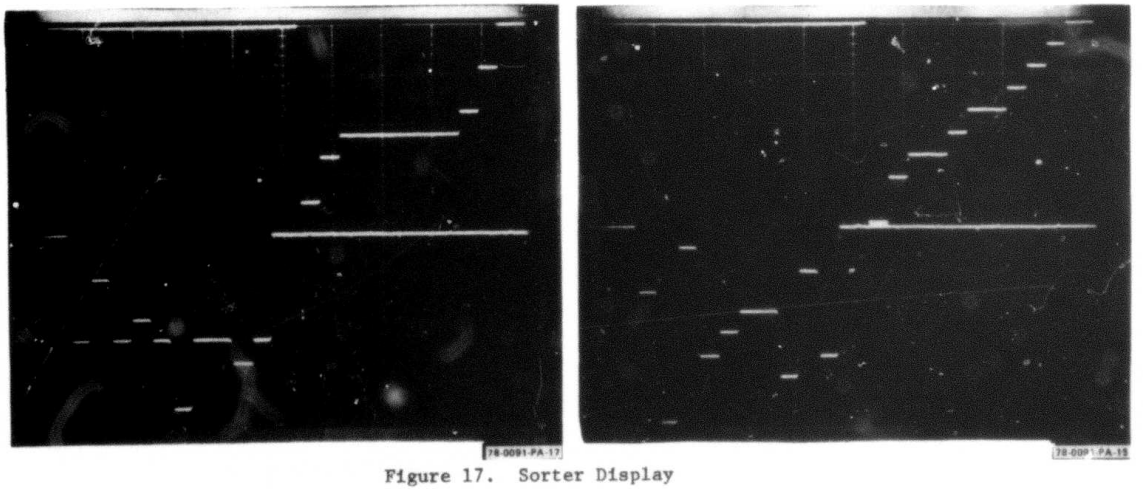


Figure 17. Sorter Display

SPOTLIGHT IMAGING OF RADAR TURNTABLE DATA

Chung Ching Chen
Harry C. Andrews

Image Processing Institute
University of Southern California
Los Angeles, California 90007

ABSTRACT

In recent years synthetic aperture radars (SAR) have proven very useful two dimensional imaging tools in various fields. Based on the synthetic aperture concepts, different imaging modes are possible with various operating characteristics. In this paper we describe a special case where the circular projection radar data are coherently processed to yield both azimuth and range resolutions. The Degree of Freedom (DOF) of such a system is derived as a means of measuring data redundancy for storage and computational requirements. The underlying radar imaging system is compared to a computer aided tomographic (CAT) system to show mathematical similaritites as well as physical differences. Experiments are performed using data obtained from the RAT SCAT radar cross section facility. Fairly good results are obtained which illustrate the versatility of coherent synthetic aperture processing of pulse to pulse high range resolution radar returns.

INTRODUCTION

In the 2-D radar imaging system, the two geometrical coordinates associated with the radar of objects are usually called range and azimuth. (In 3-D, there is one more called elevation). Range is the direction along which the signal is transmitted, reflected and received. Azimuth is the direction orthogonal to range in the surface of interest. The elevation is the direction normal to the surface of range and azimuth. The range resolution is usually obtained from timing information of the signal returns.

Depending on the requirements there are several modes of SAR: the stripping model, doppler beam sharpening mode (DBS) and the spotlight mode. In this paper a situation closely related to the spotlight mode is studied in which the relative motion between the radar and target is a circle, as in the tomography system. Unfortunately, aspect-angle-dependence of the reflectivities of the target and the shadowing effect from 3-D obscuration discourage one from applying a tomography-like reconstruction algorithm to the reflected signals. Hence, instead the SAR principles will be applied directly to small angle looks and several looks will then be registered and incoherently summed to give the full reconstruction of the object reflectivity function. A DOF as well as Nyquist rate analysis in the frequency domain will be derived to give the minimum number of data points required under specified physical constraints and requirements. Basic relations between bandwidth and resolution also will be discussed.

Finally, several experimental image results will be shown to support the theoretical work developed.

TURNTABLE DATA

In operation, the target (say a model airplane) is placed on a rotator at a distance r_0 from the radar to its rotation center as shown in Fig. 1. A reference sphere S is sitting at distances r_1 from the radar R and r_2 from the rotation center C. The angle between line RS and the target line of sight RC is α. Let (ξ,η), (x,y) be two rectangular coordinate systems with origins at C. Let (ξ,η) be associated with the target and (x,y) be with the ground of target system at an angle θ from the former coordinates, as depicted in Fig. 2. At discrete angle θ_i the radar radiates energy at a single frequency f_k. The local oscillator defined to be the reference sphere S takes the signal directly from R to S as a reference and beats the signal reflected from the target and the resultant in-phase and quadrature phase components become the data. This process continues for different f_k and θ_i to form a 2-D data array. For simplicity we shall assume that at each aspect angle the radar radiates the same set of step frequency waves, with M frequencies at the same frequency Δf. We shall also assume that the step angle $\Delta\theta$ is constant

as one advances the aspect orientation.

HYPOTHETICAL TARGET REFLECTIVITY FUNCTION

Referring to Fig. 2, let $f(\xi,\eta)$ be the reflectivity function of the target, where by reflectivity function $f(\xi,\eta)$ we mean the ratio of the received signal due to point target at (ξ,η) and the radiating signal. At wavelengths λ small compared with the curvature of the target body the target looks specular to the radar so that only those surfaces normal to the radiation path reflect strong energy back to the radar receiver. In addition, whenever a point (ξ,η) is blocked by some other points or surfaces in the line of signt to the radar, shadowing occurs. In other words, the shadowing effect occurs because of the non convexity of the surface of the target. Thus $f(\xi,\eta)$ is actually a function of aspect angle θ. Nevertheless, for ease of analysis we assume that $f(\xi,\eta)$ is independent of θ and we shall see a close resemblance of ths imaging system to that of tomography. A great deal of insight can thus be obtained by this theoretical assumption. Even if we release this assumption, as we shall do later, the DOF analysis based on fixed $f(\xi,\eta)$ is still valid in the real situation.

ACTUAL RECONSTRUCTION METHOD

Physically the radar imaging system has lots of differences from the tomography projection system because of widely different imaging characteristics and limitations.

In the radar imaging system two kinds of information are sought: range and azimuth. Range resolution is obtained by the timing of the signal return from the target point. Ideally, the relative motion between the target points and the radar should be zero to obtain range resolution with any high degree of precision desired. On the other hand, the azimuthal resolution is obtained by creating different doppler histories to different azimuthal points by way of relative motion between the target points and the radar. This seeming conflict is resolved in the Turntable system in which different frequency components at an aspect angle θ are obtained during which there is no relative motion, to give purely range information corresponding to the specific angle θ. Azimuthal information is then provided by the phase differences of the same frequency components at different θ's, which is due to the change of range of target points induced by the target motion.

In summary, the reasons we take the discrete Fourier transform on small angle data are:

1. The reflectivity function is a function of aspect angle.
2. Satisfactory phase compensation for the propagation between the radar and the target center is extremely difficult, if not impossible.
3. However, the reflectivity function can be assumed constant over small aspect angle ϕ during which the azimuth and range processing can be separated and fast FFT techniques can be employed.
4. The shadowing effect can be reduced to minimum by adopting this technique with coherent processing over small angles and then incoherent summing over large angles, as described in the next section.

EXPERIMENTAL RESULTS

Utilizing the principles outlined above actual radar returns were processed to verify the imaging potential of the I,Q components for pulse to pulse high range resolution signatures. A coherence angle of 6.4^{o} was assumed (equalling 32 pulses in azimuth) and a cross range (azimuthal) Fourier transform is taken over these pulses. The resulting range cross-range images are presented in figure 3 for various angles of rotation. The "nose", "broadside" and "tail" aspects are intuitively correct although very low quality imagery exists at this point.

To improve the image quality noncoherent integration is performed with the range cross-range images as in figure 3. With only 7 looks noncoherently summed (at 30^{o} angle intervals) the image of figure 4a results. This is a considerable improvement and clearly shows the outline of the characteristic delta wing of the F102 aircraft. By noncoherently integrating 28 looks one obtains the results of figure 4b in which a more clear image results. To investigate the degree of coherence necessary (and allowable before "range walking" occurs) figures 4c and 4d present result for 3.2^{o} coherence angles and 12.8^{o} coherence angles. In both cases the aircraft is still clearly visible although a certain amount of degradation is beginning to be apparent in both cases.

A second aircraft was imaged using the same parameters as developed above. This aircraft was an F5E and is shorter with stubby wings and wingtip pontoons. The final figure (figure 5) presents a summary of photographs for the F5E and F102 airframes for both azimuth and elevation plots. Because all parameters are fixed for these images, scales are preserved. Consequently it is clear that the F5E is a smaller aircraft and naturally has a different azimuth and elevation projection than does the F102.

CONCLUSION

This paper has attempted to present the theory of high range resolution radar imaging from both a radar systems viewpoint and a degrees of freedom or numerical analysis viewpoint. Similarity with the computer aided tomographic scanner imaging technology is pointed out. However the differences between the two systems are emphasized and a radar unique reconstruction algorithm is developed for combined coherent and noncoherent imaging. The actual reconstruction method is explained and experimental results developed to illustrate the theories presented. The pictorial images resulting from the computational procedures are surprisingly recognizable and suggest that these techniques may have some practical application in the future.

REFERENCES

[1]. M.I. Skolnik, Radar Handbook, McGraw-Hill, 1970.

[2]. A.W. Richaczek, Principles of High-Resolution Radar, McGraw-Hill, 1969.

[3]. R.O. Harger, "Synthetic Aperture Radar Systems", Academic Press, 1970.

[4]. J.C. Kirk, Jr., "A Discussion of Digital Processing in SAR", IEEE Transactions on Aeronautical and Electrical Systems, Vol. 11, No. 3, May 1975.

[5]. C.C. Chen, "Synthetic Aperture Radar and Imaging System of the Stripping mode", University of Southern California, Image Processing Institute, USCIPI Report 770, Sept. 1977.

[6]. S. Twomey, "Information Content in Remote Sensing", Applied Optics, Vol. 13, No. 4, 1976.

[7]. H.C. Marlow, et.al., "The RATSCAT Cross-Section Facility", Proceedings of the IEEE, Aug. 1965.

[8]. R.N. Bracewell and A.C. Riddle, "Inversion of Fan Beam Scans in Radio Astronomy", Astrophysics Journal, Vol. 150, 1967.

[9]. L.A. Shepp and B.F. Loga, "The Fourier Reconstruction of a Head Section," IEEE Transactions on Nuclear Science, Vol. NS-21, June 1974.

[10]. D.G. McCaughey and H.C. Andrews, "Degree of Freedom for Projection Imaging", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-25, No. 1, Feb. 1977.

[11]. G.N. Ramachandran and A.V. Lakshminarayanan, "Three Dimensional Reconstruction from Radiographs and Electron Micrographs: Application of Convolutions instead of Fourier Transforms", Proc. Nat. Acad. Sci., Vol. 68, 1971.

[12]. P.R. Smith, et.al., "Image Reconstruction from Finite Numbers of Projections", J. Phys. A.: Math., Nucl. Gen., Vol. 6, March 1973.

[13]. H.C. Andrews, and B.R. Hunt, Digital Image Restoration, Prentice-Hall, 1977.

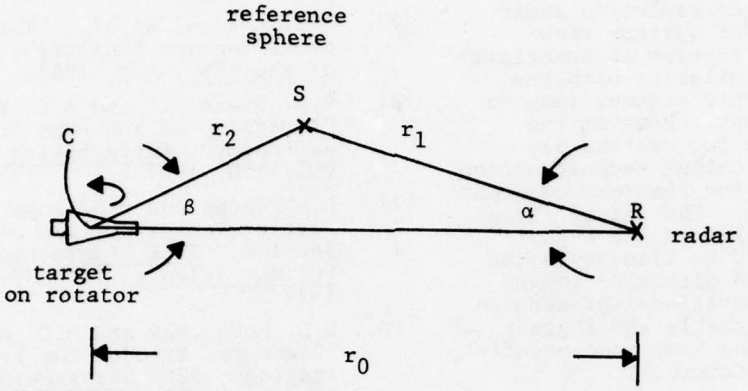


Fig. 1. relation among radar, target and reference sphere.

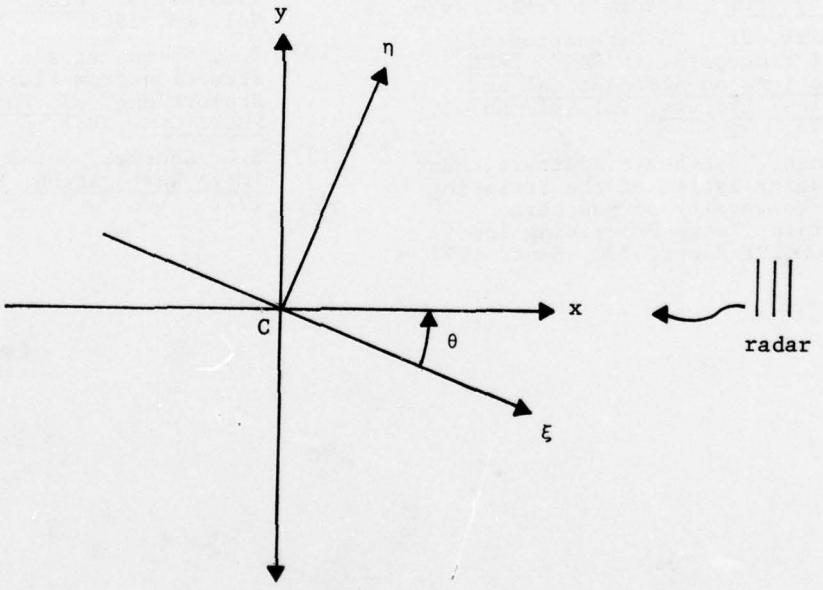
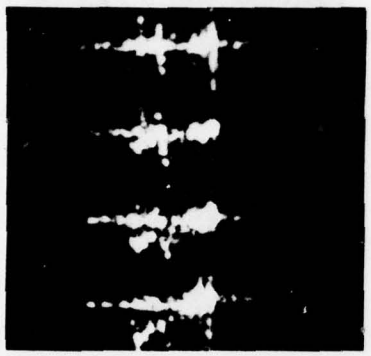


Fig. 2. geometry of two coordinate systems
(ξ,η): target (x,y): ground or radar



a) 4 looks 0^{o}-25.6^{o} (nose)

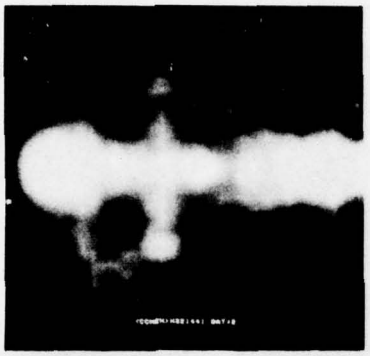
b) 4 looks 76.8^{o}-102.4^{o} (broadside)

c) 4 looks 128^{o}-153.6^{o}

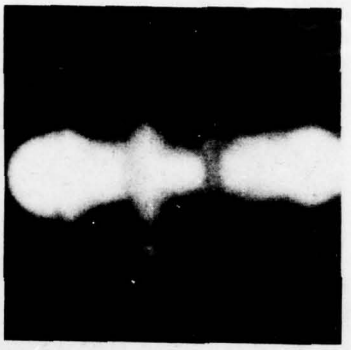
d) 4 looks 153.6^{o}-179.2^{o} (tail)

(The Radar is positioned on the right)

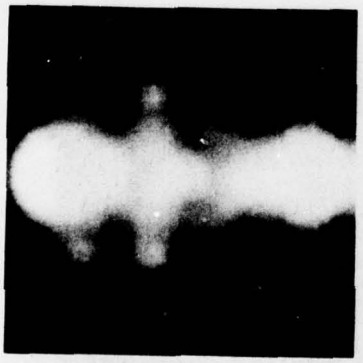
Figure 3. 6.4^{o} coherence in Azimuth at various positions of rotation.



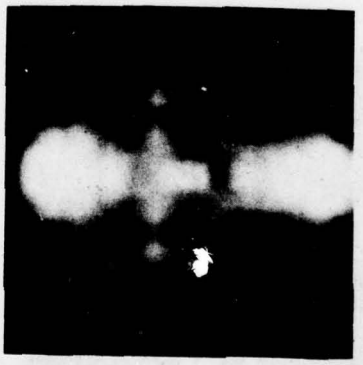
a) 7 looks (6.4^{o} coherence) spaced 30^{o} apart



b) 28 looks (6.4^{o} coherence) spaced 6.4^{o} apart



c) 56 looks (3.2^{o} coherence) spaced 3.2^{o} apart

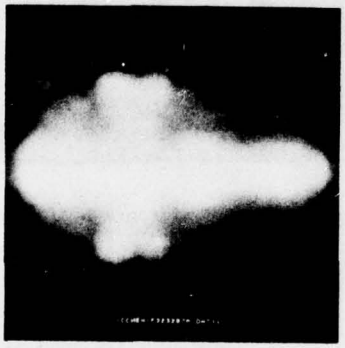


d) 14 looks (12.8^{o} coherence) spaced 12.8^{o} apart

Figure 4. F102 Imaged for various parameters

a) F102 Azimuth Image
(28 looks)

b) F102 Elevation Image
(56 looks) (1/2 scale)



c) F5E Azimuth Image
(28 looks)

d) F5E Elevation Image
(56 looks) (1/2 scale)

Figure 5. F102 and F5E Azimuth and Elevation Images (6.4^{o} Coherence)

CHARGE-COUPLED DEVICE TECHNOLOGY FOR SMART SENSORS

G.R. Nudd
P.A. Nygaard
G.D. Thurmond

Hughes Research Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265

ABSTRACT

This paper describes our continuing work[1] to design, fabricate, and test charge-coupled device (CCD) circuits for image preprocessing. Two test chips containing six processing algorithms have been fabricated and tested. The processing functions are described together with the circuit implementation and a performance evaluation.

PROCESSOR DEVELOPMENT

We have completed the design and fabrication of two test chips, as shown in Figures 1 and 2. These circuits are two-phase surface channel devices with 8 μm gate lengths. N-type silicon is used to achieve maximum speed. The algorithms implemented are

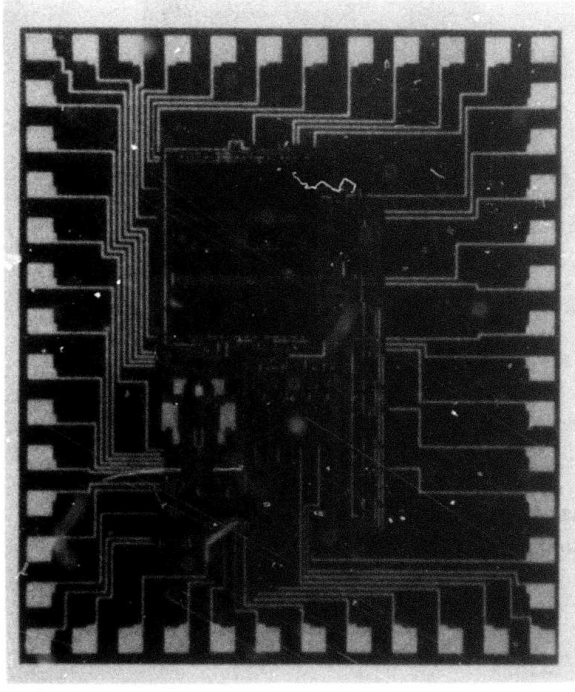


Figure 1. Photomicrograph of CCD Sobel Circuit

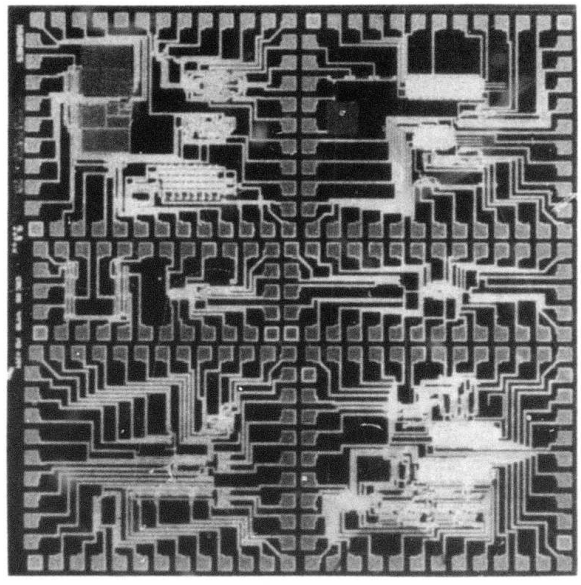


Figure 2. Photomicrograph of Test Circuit II

- Sobel edge detection:

$$f_s = 1/8 \{|(a + 2b + c) - (g + 2h + i)| + |(a + 2d + g) - (c + 2f + i)|\} = |S_x| + |S_y| \quad (1a)$$

- Local averaging:

$$f_m = 1/9 (a + b + c + d + e + f + g + h + i) \quad (2)$$

- Unsharp masking:

$$f_{usm} = (1 - \alpha)e + \alpha f_s \quad (3)$$

- Binarization:

$$f_b = \begin{cases} 1 & f_m \leq e \\ 0 & f_m > e \end{cases} \quad (4)$$

- Adaptive stretching:

$$f_{as} = \begin{cases} 2\ \min|e,\ r/2| & \text{for} \leq r/2 \\ 2\ \max|e,\ r/2,0| & \text{for} > r/2. \end{cases} \quad (5)$$

Each of these is based on a 3 x 3 array of picture elements, which are illustrated in Figure 3.

3 x 3 Array

a	b	c
d	e	f
g	h	i

Figure 3. Kernel of pixels used in the calculations, illustrating the notations used in Eqs. 1 through 5.

The first circuit, Figure 1, performs the Sobel operator detecting edges in two dimensions. The processor architecture is arranged in the form of a two-dimensional transversal filter with impulse response for the two edge components:

$$W_x = \begin{bmatrix} 1/8 & 1/4 & 1/8 \\ 0 & 0 & 0 \\ -1/8 & -1/4 & -1/8 \end{bmatrix} \quad (6a)$$

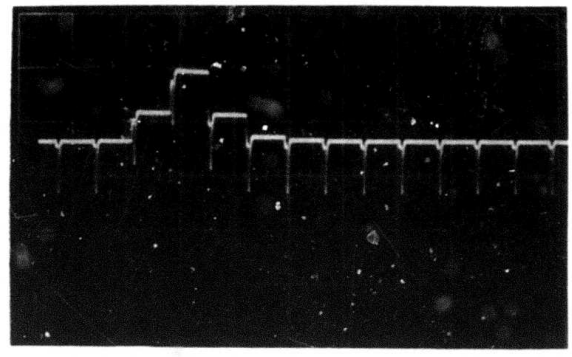
and

$$W_y = \begin{bmatrix} 1/8 & 0 & -1/8 \\ 1/4 & 0 & -1/4 \\ 1/8 & 0 & -1/8 \end{bmatrix} \quad (6b)$$

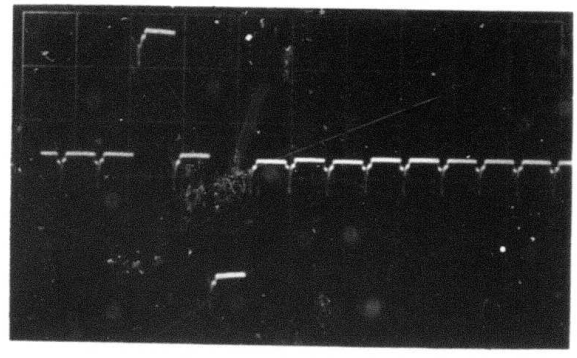
Using these two components, both the absolute magnitude of the operator (Eq. 1) and the edge direction ($\tan\theta = S_x/S_y$) is directly available. The effectiveness of the weighting techniques is shown in Figure 4. The two edge components, S_x and S_y, then are applied directly to a CCD absolute magnitude operator and a charge summer.

The performance of the CCD edge detector is illustrated in Figure 5, where an original black and white test pattern, the computer simulated Sobel (5b), and the output of the CCD processor (5c) can be compared. The clock rate for this demonstration was 15 kHz, limited primarily by the test facilities. For comparison, the CCD Sobel operation of other optical images is given in Figures 6 through 8. Our evaluations indicate that, at these clock rates, the operation has an accuracy and dynamic range equivalent to four bits.[2] We are currently unable to examine a larger gray scale because of the access time of the processed data from the commercial refresh memory we are using.

We have spent considerable effort developing a real-time processing capability to operate the CCD processor from a commercial vidicon camera, the Cohu Model No. 7120.[3] The basic data rate required

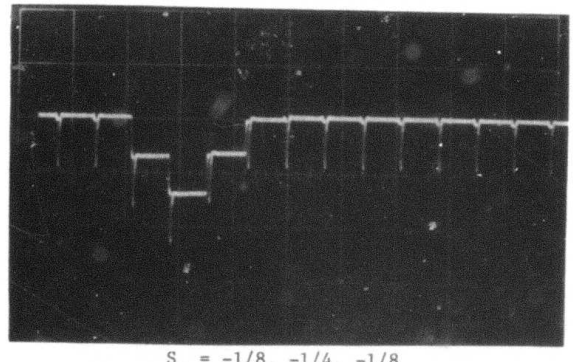


$S_x = 1/8,\ 1/4,\ 1/8$



$S_y = 1/4,\ 0,\ 1/4$

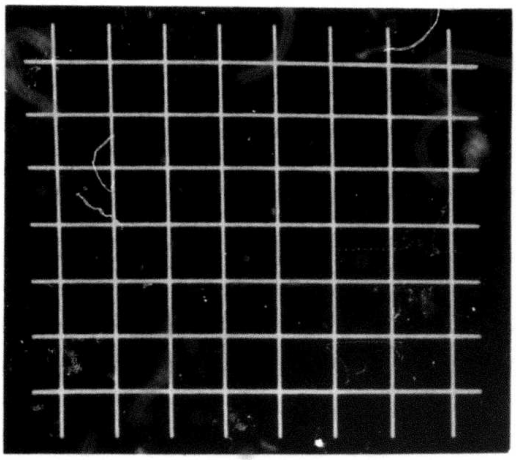
(Not to same scale)



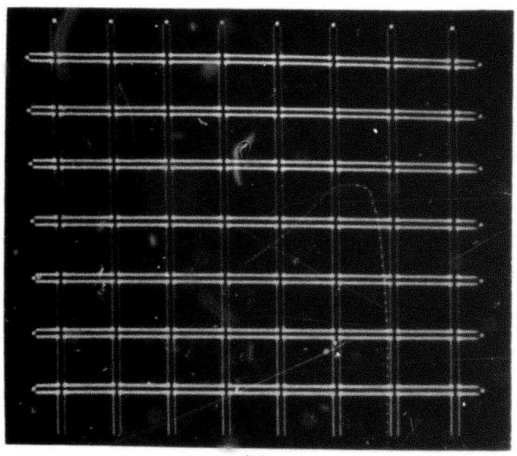
$S_y = -1/8,\ -1/4,\ -1/8$

Figure 4. Impulse Response of the 2-D Filter.

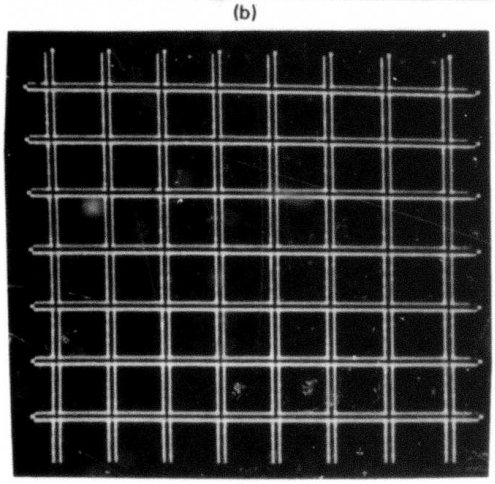
for this is approximately 7.5 MHz. We are currently operating our CCD processor at 2 MHz, which results in a slightly unsymmetrical Sobel operation, as shown in Figure 9. The frame rate is equivalent to 60 fields/sec with 512 lines, as in standard television; however, the pixel resolution in the horizontal direction is degraded by approximately a factor of three. We have tested the circuits at



(a)

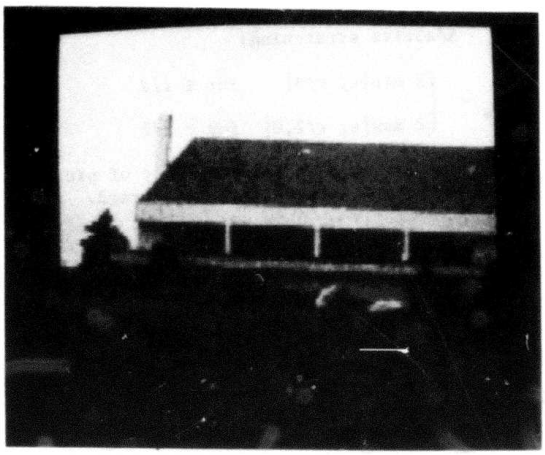


(b)

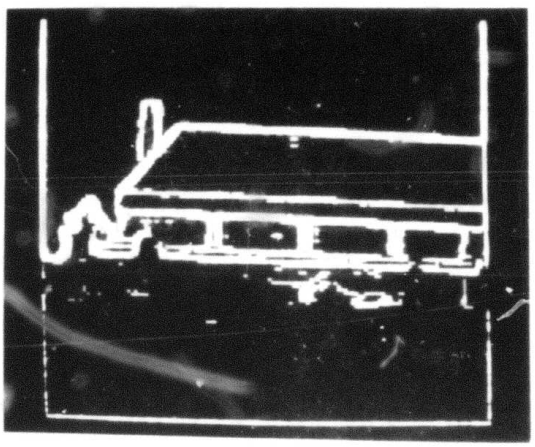


(c)

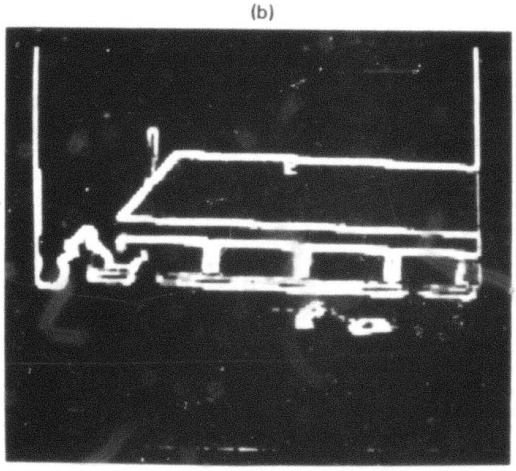
Figure 5. Example of CCD Sobel operation; (a) original, (b) computer simulation, (c) output of CCD processor.



(a)



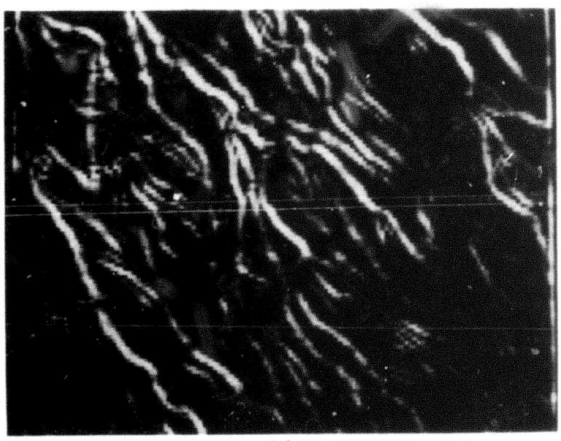
(b)



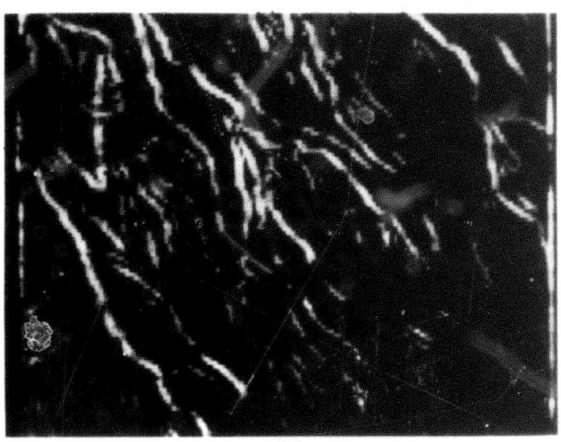
(c)

Figure 6. Example of CCD Sobel Operation on real imagery; (a) original image, (b) computed Sobel, (c) Output of CCD processor.

(a)



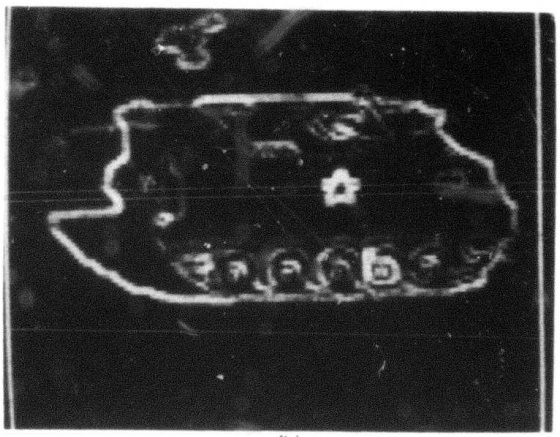
(b)



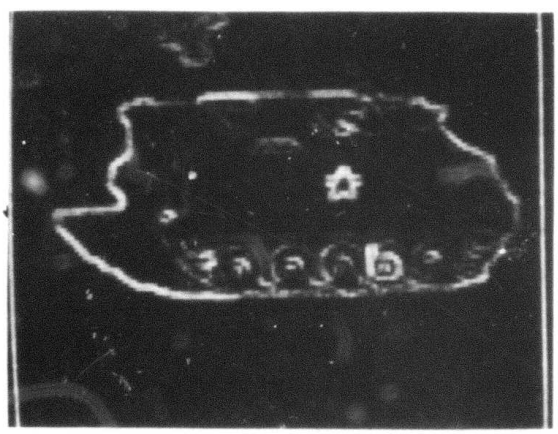
(c)

Figure 7. Example of CCD Sobel operation on real imagery; (a) original image, (b) computed Sobel, (c) output of CCD processor.

(a)

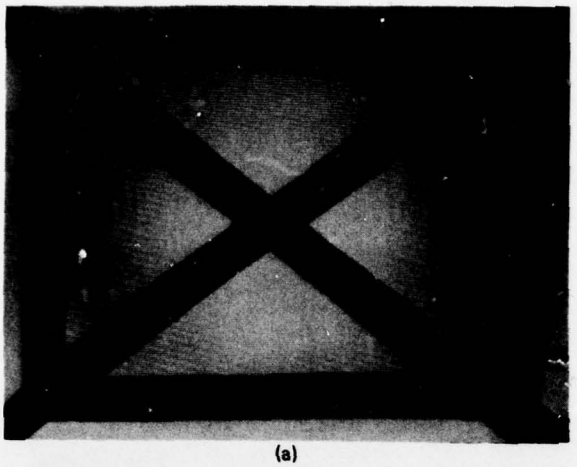


(b)

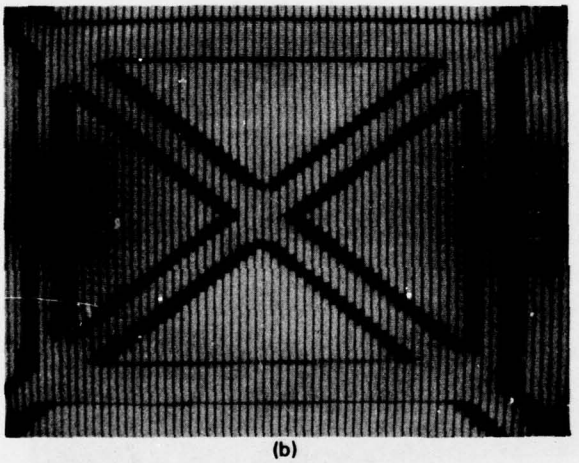


(c)

Figure 8. Example of CCD Sobel operation on real imagery; (a) original image, (b) computed Sobel, (c) output of CCD processor.



(a)



(b)

Figure 9. Example of the operation of the CCD Sobel processor operating in real time from a commercial vidicon.

these rates with a variety of images, and our intention is to increase the effective data rate in the next phase of the program to achieve truly symmetrical operation.

The edge detection circuit described above is basically an important demonstration of two-dimensional nonlinear processing. Our second test chip, which performs the operations described in Eqs. 1 through 5, is aimed at demonstrating adaptive functions based on the local mean or average. As such, the prime operators are the edge detection, local averaging, and the delayed original images. Each of the other algorithms (the unsharp masking, the binarizer, and the adaptive stretch) are arithmetic combinations of these. The original image, the Sobel, and the 3 x 3 mean derived from the second chip are illustrated in Figure 10 for a regular test pattern. Examples of the operation on a true optical image is shown in Figure 11. Each function described in Eqs. 1 through 6 (and included in Test Chip II) has been tested, and we estimate the overall performance to be equivalent to approximately 4 bits. Testing of linear combinations of the operators described in Eqs. 4 through 6 has not been completed at the full video rates, and this effort is currently proceeding. We anticipate no significant problems in this area.

NEW CONCEPT DEVELOPMENT

In addition to the above work, we have started concept development and analysis of a third test chip to perform statistics, including a 5 x 5 median filter, an analog histogrammer (including a mode and standard deviation filter, a 5 x 5 programmable processor, and several bipolar fixed filters). This work will continue into the next phase of the program when the detailed design, simulation, and initial processing will be undertaken.

DEVELOPMENT OF A REAL-TIME DEMONSTRATION UNIT

As part of our effort to interface the currently developed processors with a commercial video camera, we are pursuing the development of a small real-time demonstration unit that will include the necessary analog CCD delays, the clocks and drivers for our processor, the CCD processors themselves, and a small video display unit. This work is well under way (most of the interface circuitry has been designed), and we plan to have the complete unit available in the next phase.

CONCLUSIONS

During the previous phase of this program, we developed CCD integrated circuit processors that perform two-dimensional, nonlinear and adaptive operations at speeds in excess of two orders of magnitude higher than general-purpose computers. Our evaluations of this circuit to date indicate that it will perform as predicted[4] and can be interfaced directly to the optical sensors; this will lead directly to the development of truly smart sensors.

REFERENCES

1. G.R. Nudd, "CCD Image Processing Circuitry," University of Southern California, Semiannual Report, 31 March 1977, pp. 142-173.

2. G.R. Nudd and P.A. Nygaard, "Demonstration of a CCD Image Processor for Two-Dimensional Edge Detection," Electronics Letters, Vol. 14, No. 4, 16 February 1978, pp. 83-85.

3. "Chip Helps Detect Targets Automatically," Electronics Magazine, March 16, 1978, pp. 41-42.

4. G.R. Nudd, "CCD Image Processing Circuitry," Proc. Image Understanding Workshop, Minneapolis, Minnesota, April 1977, pp. 89-94.

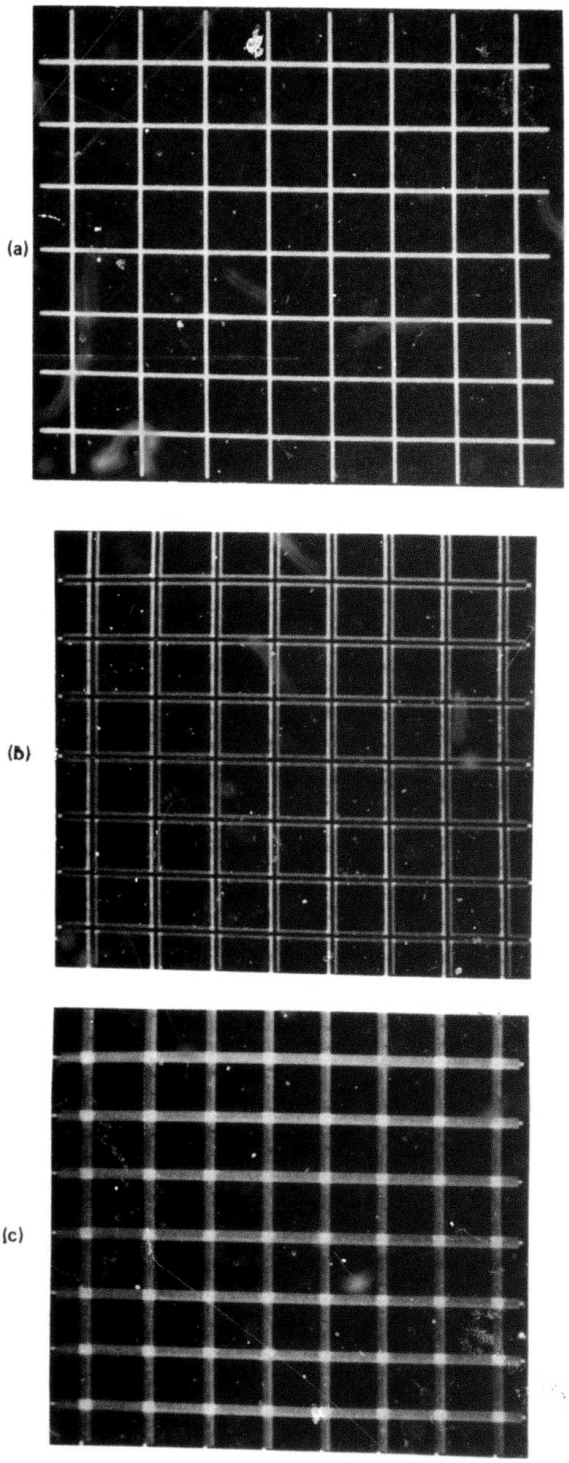


Figure 10. Output from Test Chip II; (a) original, (b), edges, (c) mean.

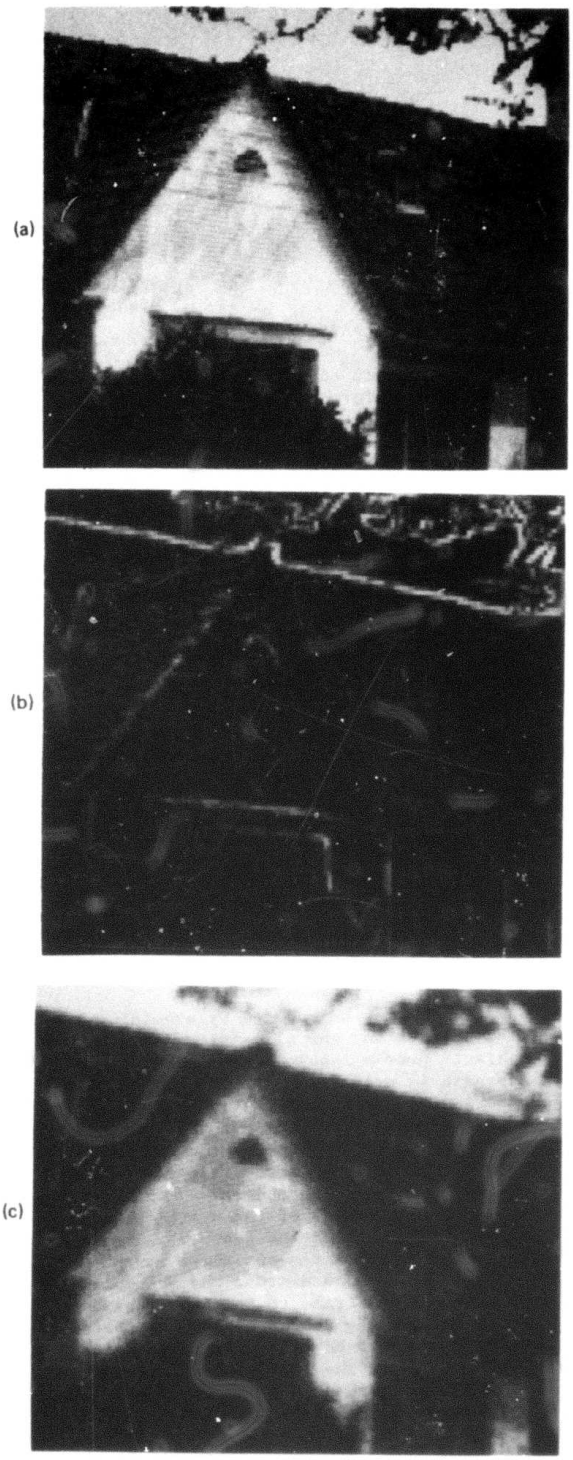


Figure 11. Example of output from Test Chip II using imagery.

TARGET SCREENER SYSTEM NOISE REMOVAL BY INTERFRAME ANALYSIS

Durga P. Panda

Honeywell Inc.
Systems and Research Center
Minneapolis, Minnesota 55413

ABSTRACT

System noise in an automatic target screener may affect the performance of the system in two ways. Firstly, the target may fail to meet the segmentation criteria of the system, resulting in a missed target. Secondly, the feature values of the segmented objects may be erroneous, resulting in missed targets as well as false alarms. Improved false alarm and detection may be achieved by accumulating information regarding the locations and the feature values of the objects from frame to frame.

INTRODUCTION

An automatic target screener system, such as Honeywell's Autoscreener, usually operates on TV compatible tactical image frames, extracts objects in a frame and optimally classifies these objects into targets and nontargets based on their statistical features. The performance of the system (probabilities of false alarm and detection) depends on the quality of data, and the image segmentor and the classifier used by the system. But the full potential of the segmentor and the classifier is often not achieved due to severe system noise.

False alarm may be reduced by examining the extracted objects and the classifier decisions on these objects over a sequence of image frames. This approach is useful and effective when noise in the screening system results in random noise in the processed image or random error in the feature values of the extracted objects, and the noise or the error is uncorrelated from frame to frame. When the image is noisy an object may fail to meet the segmentation criteria of the system resulting in a missed target. When the feature values of the extracted objects are erroneous there may be missed targets as well as false alarms. By accumulating information from one frame to the next regarding the locations and the feature values of the extracted objects improved false alarm and detection can be achieved. In the following we discuss and demonstrate this approach. In the proposed method, we first determine an interframe sequence of extracted objects containing a given candidate target in the present frame. We then determine if the classifier result on the candidate target, in the present frame, is consistent in certain manner with the classifier results on other objects, from the past frames, in the sequence. An inconsistent classifier result is modified in some prespecified manner that yields better classification result. This method of "smoothing" the classifier result consists of three distinct steps, frame alignment, interframe object matching, and decision smoothing.

FRAME ALIGNMENT

When the sensor is in motion the stationary objects in the frames will have a relative displacement with respect to the frame coordinates. To find a match for an object in a frame, a search has to be made over all the objects in the other frame. The neighborhood of search can be reduced if the two frame coordinate systems are adjusted with respect to each other to correct for the sensor motion. This adjustment is performed by the frame alignment function.

The frame alignment is based on the assumption that most of the objects in the frame are stationary. The alignment is performed by using the locational information of each object in a frame. In general, the rectification due to sensor motion may require translation, rotation, and scale change of a frame. We assume that the sensor motion between the two frames to be matched is small enough so that translation alone may give adequate frame alignement for our purposes. For example, if the frames are successive or near successive the sensor motion may be assumed to be translation only.

The method of frame alignment, called the translation histogram method, conceptually works as follows. The difference in the coordinates of an object in one frame, say F_0, and an object in the other frame, say F_1, is computed. Keeping the object in F_0 fixed, this computation is repeated for every object in F_1. This process is then repeated for all other objects in F_0. Every computed coordinate difference corresponds to a frame translation that will match an object pair in the two frames. A two-dimensional histogram of all the computed coordinate differences is made. The mode of the histogram corresponds to a frame translation that will match the largest number of object pairs in the two frames. This mode is the estimated translation necessary for the frame alignment. In order to achieve strong and robust modes the histogram is smoothed by a block filter. The frame-to-frame displacement is assumed to be

less than certain fraction, f (say 1/8th), of the frame dimensions in each of the two directions in the image. Consequently, all translations greater than f/2 (1/16th) or less than -f/2 (-1/16th) of the frame dimensions are ignored in the translation histogram.

The translation histogram method uses segmented images rather than original intensities as input. In the present context, the segmented image is a sparse binary image with zero almost everywhere and unity at the location (e.g. centroid) of each extracted object. Computing the translation histogram of two frames then precisely corresponds to cross-correlating the two corresponding sparse binary images or computing the Hamming distance between the two binary images. The mode of the histogram corresponds to the peak of the cross-correaltion.

Conventional methods of frame alignment use the intensities, the edge values or certain other property at pixel level in matching the two frames to be aligned. This may make the method sensitive to noise. Our method uses the objects extracted from the two frames in matching the frames. Noise sensitivity of the method is reduced since the chance of getting false match at object level is much smaller than that at pixel level. The data rate at object level is much smaller than that at the pixel level. These make the translation histogram method potentially much faster, cheaper, more accurate, more reliable, and more immune to noise than conventional methods.

INTERFRAME SYMBOLIC OBJECT MATCHING

A major task in interframe object matching is the selection of a suitable set of attributes or features of the objects that should be used in matching. Another major task is the matching procedure itself. Features usually used in symbolic object matching are [1,2] size, shape, color, texture, and location. The speed restriction in real time application may allow only a few and simple features to be extracted. Other considerations in extracting the features are the computational cost and the effectiveness of the features for the specific applications and image qualities in mind.

In the following we discuss Honeywell's "ordered-static-cost" method of matching the objects. The cost of matching may be of two kinds. The first, the static cost, arises due to mismatch in the features of the two objects under consideration. The second, the dynamic cost, is due to mismatch or inconsistency in the interobject structural relationships [3]. In our application the static cost is the absolute difference in the feature values of the two objects being matched. Since the objects, e.g., targets, may be moving with respect to each other, there is no constraint on the structural relationship. The only interobject constraint is that no two different objects in one frame may be matched with the same object in a second frame. This dictates the dynamic cost. Specifically, in matching the ith object in F_0 with the kth object in F_1, and matching the jth ($j \neq i$) object in F_0 with the mth object in F_1 the dynamic cost =

$$\begin{cases} 0, & \text{if } k \neq m \\ \infty, & \text{if } k = m \end{cases} \quad \text{for all object indices i and j.}$$

An optimum matching procedure should minimize the total cost of matching all objects in a frame. This may be done by computing all possible static and dynamic costs and selecting theparticular set of object matches that has the lowest total cost. However, the storage and the computational requirements are too high for this procedure. If we know the maximum distance a target may have moved between two frames, then we can restrict our search for a match to a neighborhood of corresponding size. In this regard the frame alignment helps save search time by cutting down the neighborhood size. Even then, the storage and computational requirements for finding the optimum matches for all objects in the frame may be very high. The Linear Embedding Algorithm of Fischler and Elschlager [3] is aimed at cutting down computational reauirements by trading it off with the global optimality of matching. In particular, the method may fail to find the globally optimum match if the objects with low indices in F_0 incur a high static cost when matched with their optimal object matches in F_1. The ordered-static-cost method is a similar matching procedure that is computationally more suited for our application. The procedure is independent of object indices but depends on the relative magnitudes of the static costs.

The matching procedure works as follows. Let the ith object in F_0 have K_i objects in F_1 in its neighborhood of search. We shall call these K_i object indices in F_1 the possible K_i "labels" of the ith object. The static costs for all possible labels are computed for each of the N objects in F_0. In total there are K_T different static costs, where

$$K_T = \sum_{i=1}^{N} K_i .$$

Each of these K_T costs corresponds to an object-label match. We arrange these K_T costs in increasing order. We accept at the most N of these static costs and corresponding object-label pairs as matched objects. The lowest of the K_T costs is first accepted. We then proceed to the next higher cost. If the label corresponding to this cost has already been taken by previously accepted object-label pairs, then we discard this object-label (infinite dynamic cost). If, instead, the object corresponding to this cost has already been taken, then this object-label pair has a higher static cost; hence, we discard this object-label pair and proceed to the next higher cost. If certain cost did not get discarded by the above two methods then the corresponding object-label pair is accepted as the next matching object-label pair. This process continues until all the K_T static costs are exhausted.

This algorithm will not give the globally optimum match if the static cost corresponding to an optimum object-label pair is higher than that of another object-label pair having the same label.

Consider, for example, two objects, A and B, being matched with two labels, a and b, with the following static costs:

	a	b
A	3	7
B	5	11

The object-label pairs arranged in increasing order of static cost are: Aa, Ba, Ab, and Bb. The pairs that will get accepted are Aa and Bb, even though the optimum pairs are Ab and Ba. It is possible that several iterations of a similar procedure in some suitable manner, e.g., by relaxation labelling [4] will asymptotically yield the global optimum match.

DECISION SMOOTHING

The classifier decision made on a candidate target in the present frame, F_0, may be modified based on the decisions made on the same object in the immediate past frames. Here, by "same object" we mean the object in a past frame that matches with the candidate target in F_0. The process of modifying the classifier decision in the aforesaid manner is called decision smoothing. Consider the sequence of values of a certain classifier feature of a given object from frame to frame. This sequence of values constitutes a time-series.

The error due to system noise in the feature time-series may be corrected by conventional time-series smoothing techniques. The smoothed feature values of an object may then be used to obtain a modified classifier decision on the object. A faster and simpler method of obtaining a modified classifier decision would be to treat the classifier decision itself as a binary feature time-series. One method of smoothing this binary feature is to modify the feature value in the present frame according to majority vote of the decisions on the object in the previous frames.

A problem that may be encountered in decision smoothing is an incomplete sequence. This occurs when, due to noise in the system or in the data, the segmentation method fails to extract certain object in a frame. The problem also occurs when inaccuracy in the interframe object matching process causes an object in a frame not to have any matching object in the previous frame. Thus, the binary decision time-series for the object abruptly ends at the frame when the object did not find a match. An approach to solving this problem is to skip the frame where a match was not found and proceed to finding an object match in the next frame.

EXPERIMENTAL RESULT

A sequence of five FLIR frames was processed by the Autoscreener. Figures 1a-1e show the frame sequence in increasing order of time and Figure 2a-2e shows the "objects" extracted by the Autoscreener as candidate targets. Figure 2 also shows the ground truth and the Autoscreener classifier result in every frame. The symbol "T" next to a candidate target denotes an actual target, and the symbol "C" denotes that the classifier decision was target. The test frames are numbered 1 through 5, in increasing order of time. Each frame was aligned with the previous frame using the translation histogram method. Figures 3a and 3b show the original and the smoothed translation histograms, respectively, for aligning frames 1 and 2. Table 1 shows the result of matching the candidate targets in the frame sequence by using location as the feature for matching. In the table the number following the # sign is the frame number. The table shows the object indices in the present frame, F_0, and the corresponding object numbers or labels in the previous frame. A label of "0" implies no match and an incomplete sequence.

Table 1. Interframe Object Matching

#5 F_0	#4 LABEL	#4 F_0	#3 LABEL	#3 F_0	#2 LABEL	#2 F_0	#1 LABEL
1	2	1	2	1	1	1	0
2	1	2	1	2	2	2	2
3	0	3	3	3	5	3	4
4	3	4	4	4	6	4	3
5	4	5	6	5	7	5	5
6	0	6	8	6	0	6	8
7	0	7	10	7	8	7	9
8	7	8	0	8	0	8	0
9	0	9	12	9	0	9	11
10	10	10	11	10	9	10	12
11	9	11	13	11	10	11	14
12	11	12	0	12	0	12	0
13	0	13	14	13	0	13	16
14	13	14	0	14	13	14	17
15	0	15	17	15	15	15	0
16	0	16	18	16	16	16	18
17	0	17	0	17	0	17	0
18	0	18	0	18	18	18	0
				19	19	19	0
				20	20	20	20
						21	0

The most recent frame, Frame 5, contains two objects, Object 10 and Object 12, for which the classifier decision is "target". Beginning with Frame 5, the object sequence corresponding to Object 10 is 10, 10, 11, 10, 12. The sequence is easily obtained from Table 1. The binary decision sequence corresponding to this object sequence is obtained from Figure 2 and is T, N, N, N, N, where T implies target and N implies nontarget. Thus, using majority vote on the binary decisions, the modified decision on the Object 10 in Frame 5 is N; implying that the object was a false alarm and should be classified as nontarget.

Similar object sequence for Object 12, is 12, 11, 13, 0, ?, where Object 0 in Frame 2 indicates an incomplete sequence. We now need to continue the sequence by skipping Frame 2 and finding in Frame 1 a match for Object 13 in Frame 3. Table 2 shows the result of interframe object matching between Frames 3 and 1. From this we obtain the

required object sequence as 12, 11, 13, 0, 13 and the corresponding binary decision sequence as T, N, T,?, T. Using majority vote the modified decision on Object 12 in Frame 5 is T implying that the object is a detected target.

Table 2. Object Matching in Frames 1 and 3

#3 F_0	#1 Label
1	1
2	2
3	6
4	8
5	9
6	0
7	0
8	0
9	0
10	11
11	12
12	0
13	13
14	16
15	19
16	18
17	0
18	0
19	0
20	20

DISCUSSIONS

From the above experimental results, it appears that interframe decision smoothing is potentially an effective way of combating the system noise and improving the probabilities of detection and false alarm in an automatic target screening system. The interframe object matching method may also be used in predicting the feature values (time-series) and, consequently, the signature of a target in a frame ahead in time. The predicted signature may then be used by a highly adaptive segmentation mechanism to obtain improved segmentation. The effectiveness and accuracy of the interframe analysis would depend on the features used in matching objects.

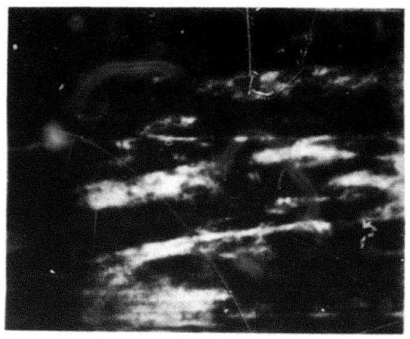
REFERENCES

1. K. Price and D. R. Reddy, "Symbolic Image Registration and Change Detection," Proceedings: Image Understanding Workshop, April 1977, pp. 28-31.

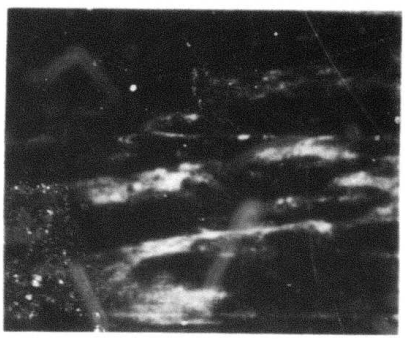
2. B. Glish, W. Kober, and G. Swanlund, "Image Registration Experiments," Ibid, pp. 32-37.

3. M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures," IEEE Transactions on Computers, Vol. C-22, No. 1, January 1973, pp. 67-92.

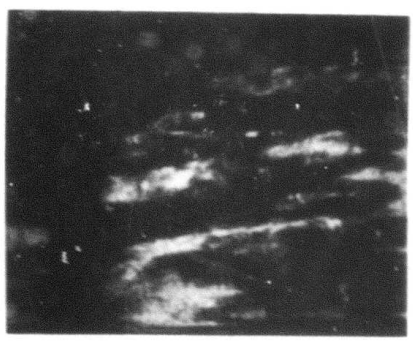
4. A. Rosenfeld, R. Hummel, and S. W. Zucker, "Scene Labelling by Relaxation Operations, "IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-6, 1976, pp. 420-433.



a.



b.



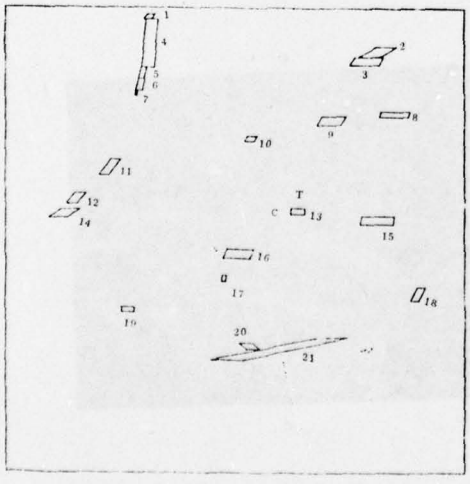
c.

Figure 1. Sequence of Five Input Frames

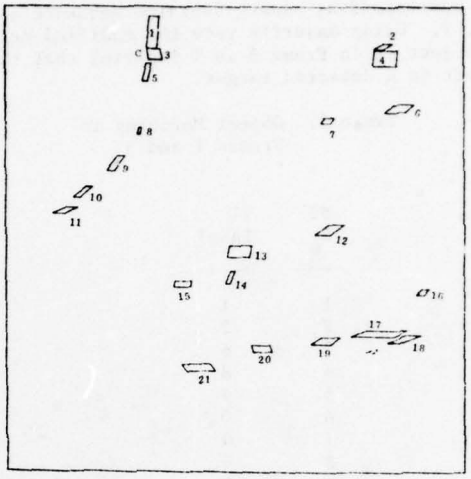
d.

e.

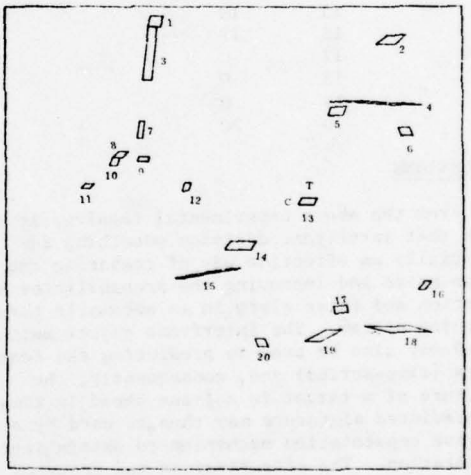
Figure 1. Sequence of Five Input Frames (Concluded)



a.

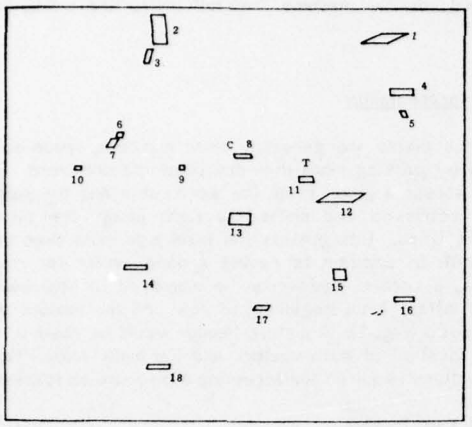


b.

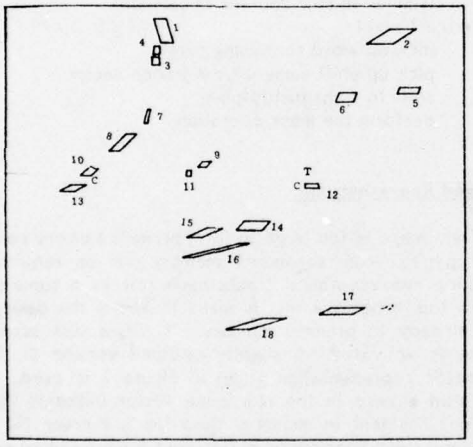


c.

Figure 2. Objects Extracted by Autoscreener from the Input Frames



d.



e.

Figure 2. Objects Extracted by Autoscreener from the Input Frames (Concluded)

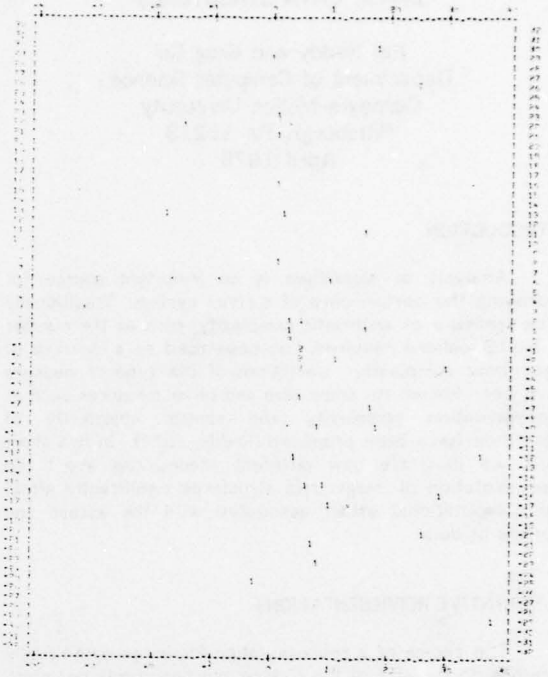


Figure 3a. Translation Histogram for Frames 1 and 2.

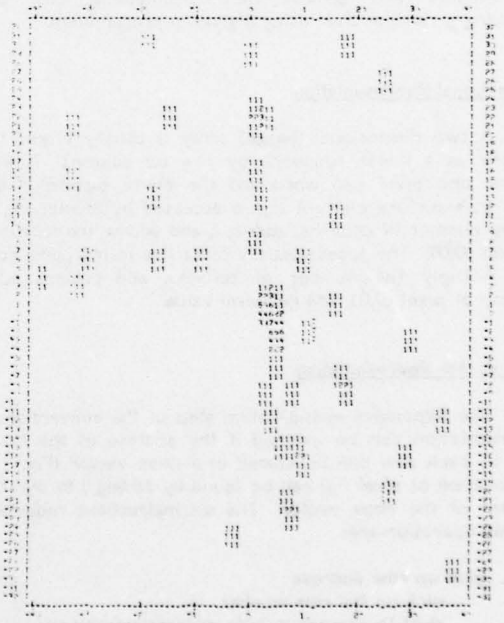


Figure 3b. Filtered Translation Histogram Corresponding to Figure 3a.

REPRESENTATION COMPLEXITY OF IMAGE DATA STRUCTURES

Raj Reddy and Greg Gill
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213
April 1978

INTRODUCTION

Analysis of algorithms is an important source for improving the performance of a given system. Traditionally some measure of arithmetic complexity, such as the number of multiplications required, has been used as a measure of algorithmic complexity. Limitations of this type of measure have been known for some time and other measures such as representation complexity and control complexity of algorithms have been proposed (Reddy, 1973). In this short note, we illustrate how different assumptions about the representation of image data structures significantly affect the computational effort associated with the access and storage of data.

ALTERNATIVE REPRESENTATIONS

The choice of a representation for image data usually depends on the size of the picture, number of bits per pixel, processor speed, and size of the primary memory. In this section we will present several alternative representations that have been used to satisfy the size and speed requirements and discuss the computational cost of accessing a random pixel using a given representation.

Conventional Representation

A two dimensional (image) array is usually stored in memory as a linear sequence by row (or column). If we assume one pixel per word and the entire picture is in memory, a picture element (i,j) is accessed by multiplying i by the number of columns, adding j, and adding the location of pixel (0,0). The access usually takes five instructions: get row, multiply the number of columns, add column, add location of pixel (0,0), and get pixel value.

Dope Vector Representation

The expensive multiplication step of the conventional representation can be avoided if the address of the first pixel in each row can be stored in a dope vector (Fig. 1). The location of pixel (i,j) can be found by adding j to the ith element of the dope vector. The six instructions required for this operation are:

1. pick up row address — 4
   - pick up the row number
   - shift to convert to byte addressing(optional)
   - add the dope vector location
   - pick up the row address
2. add j to get the pixel address — 1
3. pick up the pixel — 1

On a word address machine the shift instruction would be omitted.

Packed Representation

Since pixels are generally small numbers, space can be saved by packing more than one pixel into one word. In order to access a given pixel, the word containing the pixel must be retrieved and shifted to right justify the pixel within the word. Bits outside the pixel size must then be masked out. In addition to having a dope vector for row addresses, a second vector can be employed to hold both the word offset from beginning of row and the amount of shift required (Fig. 1). A picture header would be needed to hold the location of both vectors and the mask value. The 13 instructions required for accessing a pixel are as folows:

1. pick up header — 1
2. get the row address — 4
   (as in dope vector representation)
3. add word offset — 4
   (similar to row address calculation)
4. extract pixel — 4
   - pick up word containing pixel
   - pick up shift amount from j-dope vector
   - shift to right justify pixel
   - perform the mask operation

Row-paged Representation

If an image is too large to fit in primary memory some form of paging from secondary memory will be required. Row-paging representation treats each row as a separate page. In this scheme a test is made to see if the desired row is already in primary memory. If not, a disk access sequence is activated. A slightly modified version of the dope vector representation given in Figure 1 is used. In this version a zero in the row dope vector indicates that row is not resident in memory. Also the low-order bit is used to indicate if the row in memory has been modified and therefore must be written back onto the disk. The cost to access a pixel already in memory is only two extra instructions to the packed representation; one to test if the row is in memory, and the other to clear the low-order bit in the row address. The total cost is therefore 15 instructions, if the desired row is already in core.

Block-paged Representation

An alternate method of storing pixels would be by sub-images (blocks). The size of the sub-image block is usually chosen to be the same as the sector (or track) size on the disk. Row and column sizes of blocks are usually chosen to be powers of two. For example, a sub-image block might include eight rows and 32 words for each row. Thus a region of pixels could be read into memory without having to retrieve entire rows. The table of row addresses in Figure 1 would be replaced by a page address table as in figure 2. A zero would indicate a page not in memory and the low order bit would indicate if the page had been modified.

In general, calculating the page number for element (i,j) would require two divisions and a multiply. However, they could be replaced by shift instructions by choosing the page dimensions to be powers of two and disallowing byte sizes of three and five. This would force the number of pixels packed in one word to be a power of two. As a consequence, pages starting at the first column will be numbered from a power of 2 (e.g., as shown in Figure 2, if there were five pages across a picture, they would be numbered 0-4, 8-12, 16-20, etc). The total cost of 24 instructions breaks down as follows:

1. retrieve header	1
2. calculate page number	6
get row	
divide rows per page (or shift)	
multiply pages across picture (or shift)	
get column	
divide columns per page (or shift)	
add	
3. get page address	3
shift page number once	
add page table address	
pick up page address	
4. check for page in memory	1
5. clear modify bit	1
6. get row address	4
get row number	
modulo rows per page (mask)	
multiply by bytes across page (shift)	
add to page address	
7. get pixel address	4
(as in packed representation)	
8. get pixel	4
(as in dope vector representation)	

Hash Tables

Usually for very large pictures only a small number of pages need to be in memory at one time. Therefore, most entries in the page table (or row dope vector) would be zeros. The size of these tables can be reduced by mapping pages into a hash table modulo its length. If the hash table length were a power of 2, this operation could be done in one mask instruction.

Each non-zero table entry would point to a link list of all pages in memory which map to that table index. However, these pages should be sufficiently far from each other that the probability of any two being in memory at the same time is very small. The hash table would add 5 instructions to the cost of either row-paged or block-paged methods assuming the pixel was in the first page linked to the table entry.

The additonal instructions are:

1. save page number for comparison
2. mask the page number
3. get the corresponding entry from the hash table
4. do the compare
5. branch.

Column Calculations

The column dope vector can be completely eliminated by calculating the word offset and shift amount. Assuming the number of pixels packed per word is a power of two, the word offset can be calculated in four instructions (the same number as using the dope vector). Five instructions are required to calculate the shift amount instead of one (incrementing the dope vector). The total cost for row-paged method with hash table and column calculations would be 24 instructions. The cost for block-paged method would be 33 instructions.

1. get pixel address	4
get column	
divide by pixels per word (or shift)	
multiply by 2 (or shift)	
add to row address	
2. get shift amount	5
get column	
modulo pixels per word (mask)	
add 1	
multiply by byte size (or shift)	
subtract word size	

Subroutine Call

The instructions shown for each of these representations assume the code was written in-line. The added expense of invoking a subroutine call would be about eight instructions. Each representation would have three arguments to pass, the row and column plus the address of either the image array or a header. Putting these arguments on the stack would take three instructions. Invoking the subroutine, returning, and re-adjusting the stack would add three more. Furthermore, for all but the unpacked representations, two more instructions would be required to save and restore a register.

DISCUSSION

In the preceeding section we considered several alternative representation decisions and their computational cost. Tabel 1 shows the incremental cost of individual representation decisions. Table 2 shows the cumulative cost of increasingly complex representations for both in-line code and subroutine call. Note that it costs only 6 instructions to access a pixel from an unpacked image entirely in primary memory using an in-line macro call. This cost increases dramatically to a total of 41 instructions for a subroutine call to access a packed, block-paged image using a hash coded page table.

Conventional Representation	5
Dope Vector Representation	+1
Packed Representation	+7
Row-Paged Representation	+2
Block-Paged Representation	+11
Hash Table Representation	+5
Column Calculation	+4
Subroutine Call	+8

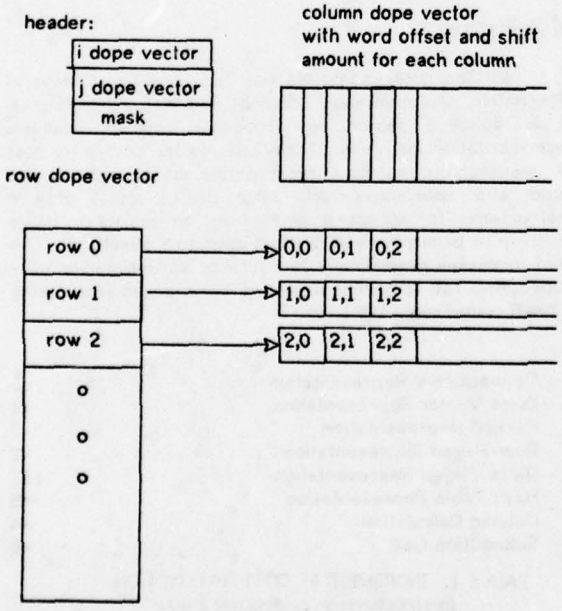
TABLE 1. INCREMENTAL COST BY FUNCTION
(instructions executed per call)

	in-line	sub-routine
Whole Picture in Memory unpacked pixels,dope vector	6	12
Whole pictures in Memory packed	13	21
Paged from Disk by Row packed	15	23
Paged from Disk by Block packed	24	32
Paged from Disk by Row packed, dope vector hashed calculate column offset	24	32
Paged from Disk by Block packed, pagetable hashed calculate column offset	33	41

TABLE 2. ACCESS BY REPRESENTATION
(instructions executed per call)

The implication of these results to image analysis can be summarized in one word: "simplify". Although a general research system must permit flexible representations to handle a wide variety of image data, a high performance operational system must use the simplest possible representation for that task and explore other architectural alternatives to random address memories such as pipe-line access or parallel array access.

FIG 1: PACKED REPRESENTATION

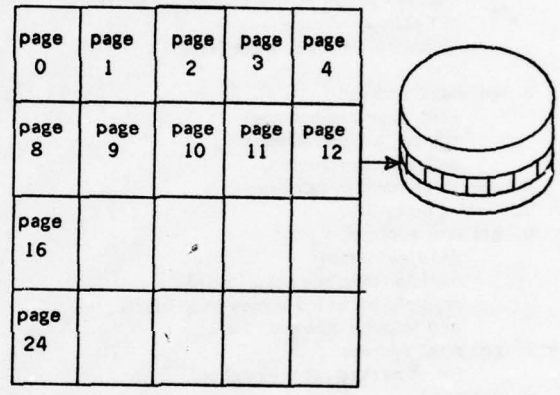


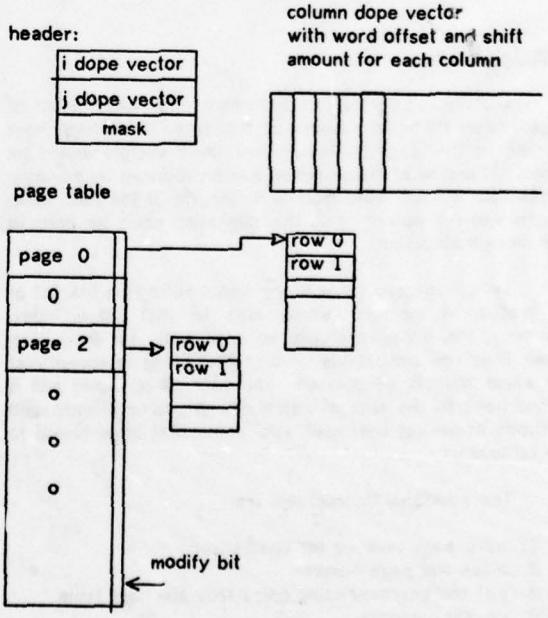
REFERENCES

Reddy, D. R. (1973). "Some Numerical Problems in AI: Implications for Complexity and Machine Architecture", in Complexity of Sequential and Parallel Numerical Algorithms (J. F. Traub, ed) Academic Press, Inc. New York.

Fig 2: block-paged representation

picture divided into pages and stored on disk





SESSION II

SYSTEMS

SYSTEMS SUPPORT FOR ADVANCED IMAGE UNDERSTANDING

J. A. Feldman

Computer Science Department
The University of Rochester
Rochester, New York 14627

ABSTRACT

As the level of sophistication in image understanding projects increases, so do the demands on the supporting languages and systems. We will describe a set of language and system facilities that have been of significant importance to our work and other DARPA efforts.

1. Introduction

There are two important reasons for investigating message-based support systems for Image Understanding. It is clearly preferable to allow distant sites to communicate about images without transmitting entire images. Appropriate conventions and protocols for this have wide-spread applications. Even within a single system, complex control and resource allocation problems arise in advanced Image Understanding tasks. The facilities described below seem to provide a uniform solution to both sets of problems.

2. PLITS, a Language for Distributed Computing

The PLITS project originally had no direct relation to distributed computing, but was concerned with developing a non-trivially new programming language. There were two basic underlying assumptions: (1) that programming languages had changed little in the previous decade despite advances in many related areas, and (2) that one could hypothesize compilers of the sophistication of the best current Artificial Intelligence programs. We began by trying to isolate the most important concepts currently available in programming systems and to see where they were compatible and incompatible. The project was called PLITS (Programming Language in the Sky) and, although it has come down a little closer to the ground, the name has stuck.

The two fundamental building blocks underlying any PLITS system are modules and messages. A module is a self-contained entity, something like a Simula or Smalltalk class, a SAIL process, or a CLU module. It is not important for the moment which programming language is used to encode the body of a module; we wish explicitly to account for the case in which various modules are coded in different languages on a variety of machines. For now, let's consider modules to be programmed in Algol-60 and also assume that there are some modules available for input, output, and file manipulation.

Modules communicate with one another solely through messages. In order to have communication, there must be something that is understood by both communicating modules. The common element in PLITS is a name which may be thought of as an uninterpreted string of characters. A message is a set of (name~value) pairs called slots. The value portion of a slot will be an element of some primitive domain (think of integers) whose representation is also generally understood.

The modules of any PLITS system will have to be able to compose, send, receive, and decompose messages. For this purpose, we must add some data types and operations to ALGOL or any other body language. In this case the primitive data types of ALGOL will have to be extended to include module and message. Each module will also contain an explicit declaration (Public) of every slot name that it can deal with along with the data type of that slot. There is a process analogous to link-editing that insures that public slot names are used consistently.

For a first example, suppose there were a module, Fibonacci, which provided the service of supplying consecutive positive Fibonacci numbers, and a module, George, which wanted to make use of this service. The code for this would be something like that shown in Example 1.

We see that George and Fibonacci both know the slot names "Object" and "Recipient" and thus can communicate. At the appropriate time, George composes a message with one slot, having as a value the system identifier for the module George itself. After sending the message to Fibonacci, this is essentially a subroutine call. The Fibonacci module simply waits for a request and fulfills it. The syntax for accessing and modifying messages treats them like the records of, e.g., Pascal.

Starting from a survey of the "powerful ideas" of programming systems, we attempted to see if there were inherent incompatibilities among them. It was immediately clear that one could not combine all the useful language primitives in a consistent way--so PLITS had to include different languages. Networking was clearly here to stay and had to be accounted for. Structured Programming seemed to be attacking the right problem with unreasonable methods. Messages were known to be a very good control primitive and were the coin of networking. The experience with RIG convinced us that messages also seemed to be a good mechanism for producing reliable yet still flexible software.

The message-module paradigm became established quickly as the fundamental solution for PLITS.

The decision to have public names as the basis of communication seems obvious in retrospect, but was difficult to arrive at. By sharing names rather than variables or sequential position in some structure, modules could be written in a way that was clear, but did not have the problems of shared storage.

It was apparent from work in automatic programming and verification that more declarative information was needed--hence we included the general notion of assertions. Although many difficult questions remain, enough clean solutions have been found to convince us that there is something fundamentally sound in the PLITS world view.

Example 1 is basically bad PLITS code; the module Fibonacci contains no error checking. Let us consider an expanded, but still weak, version which will not cause integer overflow (Example 2).

The first new notion occurs on line 4, where a public slot name of type "problem_type" is declared. The type problem_type is a fixed sequence of uninterpreted symbols exactly like the Pascal "enumeration" type. There will be several public enumerations in a PLITS system. In lines 9-11, a prepackaged message is assembled and stored in the message variable, My_Complaint. The other new code is in lines 21-27; the Recipient slot of My_Complaint is filled in from the Request. If there is a Complaint_Dept slot in this request, the module which is its value will be sent the complaint. Otherwise, some default complaint handler, City_Hall, will hear about it. The name of the Recipient module (which may have been awaiting an answer) is passed along to the Complaint_Dept, because there might be some appropriate response to the problem. For example, there could be some double precision Fibonacci module which would be able to return an appropriate value if George were prepared to accept it. This would require that George handle double size integers; that is not hard to arrange, for example by an extra slot for the high order part.

There is a more interesting problem in the control discipline used in the coding of the module George given in Example 1. The statement on line 8 is:

Mess2 ← Receive from Fibonacci.

But we saw in the expanded Fibonacci module of Example 2 that there might be an error recovery module that would supply the answer if Fibonacci could not. The coding style of line 8 requires that the answer be conveyed back to Fibonacci and then to George, but there is nothing to be gained by retracing our steps. To solve this and a number of other control problems, we will add one more construct, transaction, to PLITS. Intuitively, a transaction is a key which can be used in the regulation of message traffic. We could replace 8 with

8' Mess2 ← Receive about Key4 ;

where Key4 is a transaction which is identified with the generation of this sequence of Fibonacci numbers. Selective receives based on transaction keys allow a receiving module to be programmed without regard to which module will ultimately send it the message. Yet the receiving module is still able to keep separate "conversations" distinct.

3. DSYS--A Distributed System

With the PLITS style of programming as background and a source of examples, we are developing an experimental system (DSYS) to support high-level distributed computing. DSYS will run on the seven computers in our laboratory: four ALTOs, two Eclipses, and a PDP/10. It will provide facilities for defining and running PLITS distributed jobs (DJOBs).

Even on a single machine, there will have to be some underlying programs which handle messages. We will call this collection of programs the Kernel for a PLITS site. The Kernel is a conventional multi-programming monitor which sequences through the modules on its "ready" queue. The Kernel also maintains data structures describing modules which are "suspended" waiting to Receive a message of a specified sort. These data structures, together with analogous ones for messages which result from Send statements, suffice to implement the PLITS message primitives.

A problem arises if the modules are written in different body languages. It may be the case that languages differ in their representation of primitive data types (e.g., real). We require that the representation of primitive data types be uniform within a site. This, as well as other considerations, may give rise to the situation where there is more than one site on a given machine involved in an individual distributed job (Djob). Figure 1 is a graphic representation of the breakdown of

```
1  Begin "George"
2    Public Integer Object;
3    Public Module Recipient;

4    Begin Comment George's thing;
5      Integer I,J,Next_Fib;
6      Message Mess1, Mess2;
         .
         .
         .
7      Send {Recipient~Me} to Fibonacci;

8      Receive Mess2 from Fibonacci;

9      Next_Fib←Mess2.Object;
         .
         .
         .
10   End
11 End "George"
```

```
Begin "Fibonacci"
  Public Integer Object;
  Public Module Recipient;

  Message Request;
    Integer This, Last, Previous;
      Last←0; This←1;

  While true do
    Begin
      Receive Request

      Previous←Last;
      Last←This;
      This←Last+Previous;

      Send {Object~This} to Request·Recipient;

    End
End "Fibonacci"
```

Example 1

```
1  Begin "Fibonacci"
2    Public integer Object;
3    Public module Recipient, Complaint_Dept, Complainer;
4    Public problem_type Problem;

5    message Request, My_Complaint;
6    module Complainee;
7    integer This, Last, Previous, Biggest;

8    Last←0; This←1; Biggest←2^31 -1;

9      My_Complaint←{Problem~Overflow,
10                   Complainer~Me
11                  }

12   While True do
13     Begin
14       Receive Request
15       Previous←Last;
16       Last←This;

17       If Biggest - Last > Previous
18       then Begin This←Last+Previous;
19            Send {Object~This} to Request·Recipient
20            End
21       else Begin
22            Put (Recipient~Request·Recipient) in My_Complaint;
23            Complainee←If Present Request·Complaint_Dept then
                                    Request·Complaint_Dept else City_Hall;

24            Send My_Complaint to Complainee
25            End
26     End While Loop
27 End"Fibonacci"
```

Example 2

functions and terminology which we have adopted. It is convenient to divide the PLITS support functions into two subsets carried out by the site kernel and by the DSYS Host Control Program (DHCP) respectively. In the example, there are two Djobs, A and B, which have no connection but happen to be both distributed over Machines 1 and 2. Djob A consists of three sites: S11 and S12 on Machine 1 and S21 on Machine 2. Each site has a kernel associated with it as described above. The kernel performs the following functions:

(1) distributes messages within the site;
(2) forwards messages to and from other sites;
(3) carries out needed representation shifts for inter-site messages;
(4) allocates resources within the site;
(5) generates unique (world-wide) names;
(6) checks for errors and assertion violations.

We have briefly discussed the first three functions. The fourth function, resource allocation within the site, is concerned with storage allocation and reclamation, scheduling of ready modules, etc. The fifth function is the generation of unique names for modules and transaction keys. Error and assertion checking is discussed below.

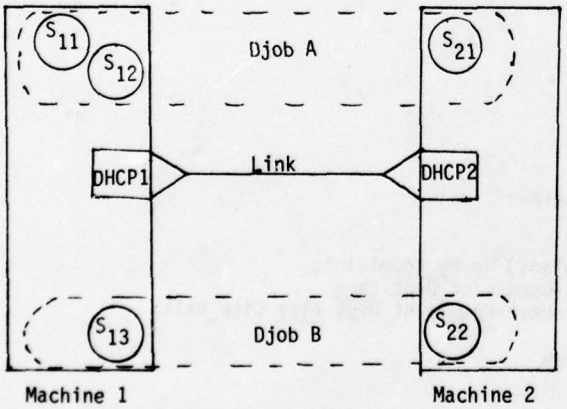


Figure 1

Each DHCP is an extension of its machine's operating system. It performs four main functions:

(1) distributes messages among sites local to this machine;
(2) forwards messages to and from other machines;
(3) starts and stops Djobs, and provides access to other operating system services;
(4) checks for errors and assertion violations.

Let us first consider the problem of setting up a Djob. If there are two sites on the same machine with the same representations, the DHCP only has to check that the use of public slot names is compatible -- essentially the same process as combining the externals of two load modules. If there are several machines involved and there is an incompatibility in representation of a primitive data type, then some conversion routines will have to be automatically invoked. The ARPA network voice protocol presents a good model of a scheme in which a dialogue between machines is used to reconcile representation differences before messages containing data are sent. All of this is fairly messy, but should only be necessary when a new PLITS language processor is brought up on a machine. In the usual case, the standard conversions between sites will have been established and the negotiations between machines will be simple.

When a PLITS message is sent by a module in a site, its destination is checked. If it is within a site, the site kernel handles it; if not, it is given to the local DHCP. If the destination is within another site on the same machine, it is given to the kernel for that site; if not, the DHCP has it forwarded to the appropriate machine. This is the job of DHCP functions 1 and 2 above. To do this effectively requires quite a lot of mechanism beneath the surface. Problems faced include reliable transmission, flow control, error handling, and providing user services in a distributed operating system.

The present DSYS design provides "emergency" messages as the mechanism that the system uses to report asynchronous errors to a module. If a module has an emergency message on its input queue, the system will include a notice that there is a pending emergency message as part of the normal response to any call that sends or receives a message. This is only an initial attempt at providing a uniform mechanism for errors and other asynchronous conditions.

An experimental version of DSYS is up and working in our local network. There are experimental DHCP's for the ALTOs and for the PDP-10, and the

Eclipse DHCP is in the final stages of debugging. Each DHCP has most of a Communications Manager, a name server, and a rudimentary Job Manager (presently a Request Fielder that provides file service).

There is a rapidly growing awareness [Hoare 77] that the paradigm of a collection of communicating sequential processes is a useful and powerful concept for solving problems and for developing computer systems. In the usual way, progress requires the development of concrete systems which both test ideas and lead to new ones.

Our work on DSYS is motivated by the requirements of PLITS and by our experience with RIG. Our desire to provide flexible communication facilities for user jobs in a distributed operating system has led us to take a fresh look at some of the problems of distributed computing. In particular, we are developing a scheme that provides both a uniform user view of inter-module communication and a flexible system view of resource management.

Further, we are developing the idea of a distributed user job, and designing mechanisms for handling errors and exceptional conditions in distributed systems. At the low level, we are working on communication protocols that use end-to-end flow control and reliable transmission, allow fine control over buffer space allotments for arriving messages, and provide detailed feedback for intelligent flow control when such information is available.

To help guide the work on DSYS, we find it useful to express design goals as questions. The present collection of such questions is outlined below.

What kind of a system is required to support a programming methodology in which sequential processes ("modules") communicate via messages? How can such a system be designed to present a uniform user view of intermodule communication, independent of whether the communicating modules run on the same computer?

What can be done to provide systematic conventions for dealing with the errors and exceptional conditions that occur in distributed computations? In particular, how can such a system be made robust? What can be done to maintain the integrity of a distributed system (and of innocent user jobs) when either a user job or a part of the system fails?

How should "user job" be defined? What services should the distributed system provide, and how should user jobs deal with the distributed system? What are the special problems of user jobs in such an environment, and how can the distributed system help?

How can performance be monitored and distributed computations (and systems) tuned? In general, how should the programmer think about an execution of his computation? What tools can the system provide to help in this regard? Such tools should also be helpful to the system designer.

How can such a system be made reliable? Are there practical descriptive techniques for the protocols of real distributed computations? How can such a description be used effectively to uncover design problems or generate tests? How much of this can be automated?

A MODEL BASED VISION SYSTEM

Rodney A. Brooks, Russell Greiner, and Thomas O. Binford

Artificial Intelligence Laboratory, Computer Science Department
Stanford University, Stanford, California 94305

Abstract

We describe a partially implemented system called ACRONYM which is designed to recognize instances of generic and specific models in photographs. The system is being built with airfields, oiltanks and aircraft as examples. It is intended to be easily extended to other objects. This interactive system will be taught by photo interpretation experts to locate specific classes of objects. The user communicates with the system in terms of object models. The system has a high level language for building object models with graphics support for the user. To make use of the models in a general way, the system derives descriptions of observable properties from three dimensional models and matches these against the image in a relaxation process.

Introduction

This research addresses the problems of identifying objects based on generic descriptions, and of providing tools for users to specify vision tasks in a natural way. In a typical scenario, a photointerpreter will give a brief symbolic description of a typical airfield, and describe some specific airfields. He will show some examples of airfields, from which both specific and generic properties will be inferred. The system can infer statistical distributions, but that is not very interesting. It is now reasonable to expect only simple inferences. Rich inference of generic properties depends on broad world knowledge. For example, to infer reasonably about lengths of runways requires knowledge about their function for takeoff and landing of aircraft, and the distance required for these operations.

Objects are modeled in a high level language based on a "generalized cone" representation of objects. The representations of most objects are very compact; they are segmented into volume elements which seem quite natural to the user. This geometric language provides graphic aids for the user for modeling generic objects and scenes, as well as specific instances.

For a specific task, an Observability Graph is determined which contains task-specific and quasi-invariant observables and relations. Task-specific information is based on knowledge such as sun angle and camera position. Observables are those features and relations which are detectable, i.e. that are easily found by operators; they are expected to have reasonable contrast and be large enough to find. Quasi-invariants are those features which remain nearly invariant over a large range of viewing angles.

The program matches these models against images which have been processed from the pixel level to higher level ribbon primitives. The data is thus already structured into natural components. Matching is carried out by a relaxation process. The conditions which go into detailed verification vary enormously in their cost and effectiveness. A general structuring of the matching process into coarse and detailed phases reflects an ordering of priorities.

The system is being implemented in MACLISP.

The Knowledge Base

The model base for this system has a variety of sources and uses multiple interconnected representations. The primary representation is in terms of three dimensional generalized cones (Binford 1971) for volume elements. Logically there are three principal graphs; the three dimensional Object graph, the two dimensional Appearance graph and the Observability graph which has 2d and 3d features. The contents of the latter two of these graphs are derived from the first, and may change over the course of recognition and display tasks. One of the most important features of these derived graphs is that they always contain back pointers to the object graph (and possibly to each other) so that routines can refer back to the original three dimensional model. See figure 1.

The Object graph contains both generic and specific hierarchical models. At the highest level there are SCENES (for instance an airport). SCENES are made up of OBJECTS (e.g. airplanes, oil tanks or runways) with spatial inter-relationships. OBJECTS are graphs (usually trees) of attached PARTS, which are graphs whose primitives are represented as generalized cones. Both generic descriptions of scenes and objects, and detailed descriptions of specific instances of them, are included in these graphs. Scenes and objects are both grouped into classes, such as airport-scenes, airplanes, oil tanks etc. Properties common to members of these classes can be given specifically in a high level description language, or can be inferred from specific examples already included in the object graph. Specific examples can be specified in the same high level language, either by complete description or by making more specific the properties of the general model. Eventually, instances extracted from processed images may also be incorporated into the object graph. This will be useful as a method for initially training the system, and as a means for the system to become more familiar with particular classes of scenes

or objects through experience.

As mentioned above, OBJECTS are represented as graphs with PARTS at the nodes. PARTS are subgraphs whose primitives are single generalized cones. A generalized cone representation for three dimensional objects was first described by Binford (1971). Restricted versions of this representation have been used by Agin (1973) (circular cross sections) and Nevatia and Binford (1977) in visual recognition systems. Marr and Nishihara (1976) restrict themselves to circular cross sections, straight spines and constant sweeping rules to determine the spatial relationship of an object and an observer. Miyamoto and Binford (1975) have used polygonal cross sections, straight spines and piecewise linear sweeping rules for object modeling.

As in Marr and Nishihara (1976) and Miyamoto and Binford (1975), each cone has its own coordinate system and the arcs of the OBJECT graph are transformations between the coordinate systems at each node. In our representation not all arcs require an explicit coordinate transform (the default is the identity). Eventually we may want to include other information in the OBJECT graph such as explicit mention that the OBJECT is symmetric about some plane.

Single PARTS have a cross section, a spine and a sweeping rule. The cross section is swept along usually perpendicular to the spine, a space curve. The cross section varies according to the sweeping rule and thus defines a three dimensional volume. Certain minimal conditions on the three descriptors of a PART have been assumed throughout the code written so far. We have not yet implemented the full generality which these conditions permit. Incremental additions to the code can push towards that level of generality without any obvious impediment. These assumed conditions are given in the following paragraph.

The spine is a continuous space curve parameterized between zero and one ($0 \leq s \leq 1$), with continuous tangent function. In the canonical coordinate system assumed for a PART, the spine starts in the positive x direction from the origin with the tangent lying along the x-axis. For each value of the spine parameter we need to calculate the orientation of the plane normal to the spine. This may be done explicitly by some function associated with a particular spine or implicitly by say, a declaration that the spine is a straight line. The cross section is defined in the y-z plane, i.e. at $s=0$. In its most general form a cross section is a collection of 2-d specializations of generalized cones, called ribbons, each labeled as positive or negative. One can think of all the positive ribbons being pasted together in their correct positions and the negative ones are cut out of the area defined by the positive ones. Thus a cross section can have many regions, perhaps with holes in them. In the same way that 3-d generalized cones more natuarally represent volumes than do the surfaces which enclose them, so do 2-d ribbons represent an area more naturally then does a list of line segments. For the applications we are currently considering a single ribbon defining a simple area without holes should suffice. Usually cross sections are described in our system by special case terms such as circle (which does not readily fit a ribbon description), square, rectangle etc. The sweeping rule must be defined for each value of the spine parameter as a two dimensional linear transformation (note that this does not mean the transformations are linear in the spine parameter). Thus the cross section at any point along the spine can be calculated by applying the sweeping rule to the cross section at $s=0$, followed by the rotation given by the spine.

Returning now to the high level input language, models of scenes, objects and parts can be named and described in the input language, edited interactively with display graphics (if the model is sufficiently explicit to fully specify an appearance graph) and stored in data bases along with derived observability information (to be described later). Some examples of the current version of the input language are shown in figure 2. Tree structures can be naturally described using nesting of S-expressions within an object description. The coordinate transforms can be specified by including "with position" and 'with rotation" clauses. The outputs of the parser are the tree structures at the various levels of description, with those slots which have had values specified filled in at the nodes. Any item can be given an optional name and later be referred to in one of two ways. If referred to simply by name a pointer directly to the item is used as the tree node or slot filler. If referred to in a "just-like" phrase, a copy of the item is made using the same slot fillers and sub-trees the original used. By specifying further qualifications of this copy the contents of specific slots can be altered while retaining most of the structure derived from the prototype. In many places (such as position specification) the parser uses the LISP EVAL function to allow the use of arbitrary S-expressions and bound variables.

Generic descriptions of a scene, for example an airfield with the ground plane in the x-y plane of the coordinate system, can be input in this specification language. Descriptions can include a description of the range of the number of each type of object to be found in a typical airfield scene, along with generic descriptions of those objects and their component parts. For instance a generic description of an OIL-TANK is given in fig 2a. When describing a specific scene this can be used as a prototype if desired as in fig 2b. This object specification would then appear nested in some description of a scene. The position value gives the position of the object coordinates relative to the scene coordinates.. A rotation specification for the whole object could also be included, but for this example the necessary rotation to transform from the canonical coordinate system of generalized cones to the scene coordinates has been inherited by the single part from the prototype part TANK-BODY. New copies of the cross section and spine are made using the "just-like" construct as the numeric values need to be made specific. The other slots of these two specifications are inherited from the prototype, but since in this case they are already completely specific, no modifications need be made.

Rather than input a generic description it can be inferred from examples in the manner of Winston (1975). Between these extremes some properties can be described directly by the user, while others can be inferred by the program from its known examples. It is not yet clear exactly when these inferences should be made by the program and so far this decision is not made automatically but only when the user specifically invokes the necessary functions. It is intended that this question be examined in much more detail.

The Appearance graph has a variety of possible uses. Since this is intended to be an interactive system for photo-interpretation it is an important advantage to maintain a representation which is intuitively natural for the user. The Appearance graph is used to produce a two dimensional image for display to the user during the model building and learning phases (fig 3 for example). This gives the user some feedback from the model building process; she can see what the data base

thinks the models look like. Similarly when the program has built up a three dimensional model from an actual image the user can get a much clearer idea of what the program is "seeing" by looking at a display picture generated from that cone representation. In the current system this is the only function of the Appearance graph.

Baumgart (1974) suggested that such a graph could be used to produce a synthetic image which would be matched at the pixel level against images normalized to a standard point of view using recursive windowing techniques. Thus an area could be monitored by comparing new pictures against a synthetic noise free image. When a picture is examined which has significant differences the information about what part of the three dimensional model is no longer valid will be extracted from back pointers attached to the Appearance graph. The model based recognition system can then be invoked to decide what has been added or removed from the monitored site.

This graph might also be used to extract observable features which are dependent on a particular camera and sun position, such as occlusion and shadow information. It would be used in this guise for features which do not have the invariance with respect to camera and sun that is common to the features extracted from the three dimensional graphs.

An Appearance graph can be produced from any scene whose graph arcs, nodes and value slots all have specific values. The surfaces of the objects and their positions in space must be extracted from the generalized cone description. These can then be converted to camera coordinates and projected onto a plane. It must be decided for each surface whether it faces the camera and so is potentially visible, and if so whether it is obscured by other surfaces. The initial culling of surfaces, i.e. discarding those which wholly face away from the camera, has been implemented for planar surfaces and a small class of curved surfaces. The discussion that follows describes how to produce the appearance graph for a single convex part. It seems clear how to extend many of these techniques to handle a class of non-convex parts but it is not yet clear whether it is necessary to do so for the domain of images which are being investigated. Thus far we have stressed building up other capabilities, but it is intended to extend the hidden surface algorithms to handle parts partially or wholly occluded by others.

Most traditional hidden surface algorithms rely on the fact that objects have polygonal surface representations and the surfaces are planar; e.g. see the survey of Sutherland, Sproull and Schumaker (1973). Braid (1973) includes sections of elliptical cylinders but relies on special case solutions for pairs of six primitive volume elements. Extensions to more general curved surfaces would not fit easily into his system. In the system to be described here, the surfaces are extracted from the generalized cone representations of the parts. In general these surfaces are not planar polygons and they are not approximated by such. The task of producing the outlines of each surface and then the back surface culling technique used will be described.

The cross section is produced for spine parameter s=0 in terms of two dimensional curve primitives. The cross section slot in a part description points to one or more cross section descriptions. These are handled independently and then merged to produce a cross section, possibly non-convex and with holes in it. For instance a cross shaped cross section might be described by two elongated rectangles at right angles to each other with a common center. The merging process would eliminate the internal lines leaving the outline of a cross. For the current applications this generality seems unnecessary and so the merging is not done. The cross sections are usually limited to a single cross section description. The "type" slot in the cross section description of a cone is used to invoke a function of the same name, which uses the rest of the slots as arguments. New routines which produce cross sections can be added independently of the rest of the code as long as they describe the cross section with those primitives supported by the rest of the program. The implementation currently includes straight line segments, circular segments and (partially) elliptical segments as two dimensional curve primitives. To introduce a new primitive cross section element it is only necessary to include a few functions to handle such things as the final drawing stage, the production of the corresponding surface element when it is swept out by the sweeping rule and the rules to translate, rotate and deform it using a two dimensional linear transformation. A circular segment for instance is represented as a center, a radius, the orientation of the plane in which it lies and two angles which delimit the end points of the segment. This is the representation used throughout the production of the appearance graph and in that graph itself. It is not approximated by straight line segments until it comes time to place a line drawing in some output buffer. The program then calculates the size of the image and based on the output device (e.g. display terminal or Xerox Graphics Printer) decides how many straight line segments to use. The cross section routines also have to mark which points in the cross section will be swept out as visible lines along the sides of the generalized cones.

Logically the information provided by the spine and the sweeping rule could be obtained independently of each other so that new spine and sweeping rule types can be added independently. New types can be added in the current implementation without regard to their interactions but some of the pairings of simple cases are handled specially when substantial computation savings can be made. In the general case each given type of spine provides a position vector and orientation matrix for any requested value of the spine parameter. The sweeping rule provides a two dimensional linear transformation for any given value of the spine parameter. These two transformations are combined to give a single transformation which (for spine parameter s=1) can be used on the whole cross section to get the face at the other end of the generalized cone, or (with intermediate spine parameter values) it can be used on the sweeping points to locate points on lines which lie on the swept surfaces of the cone. However in the case of a straight spine and a constant sweeping rule, this transformation is merely a translation and all lines swept out are straight lines. Thus considerable savings in the number of arithmetic operations can be made for such a simple case. In the domain being investigated many objects can be modeled using precisely these simple cases.

When carrying out the back surface culling it is not as clear how it should be done independently of the combination of the primitives for cross section, spine and sweeping rule which were used in producing a particular surface. So far the culling routines have been implemented only for planar faces, and surfaces where a circular segment has been swept along a straight spine with either a constant or linear sweeping rule. For the case of a planar surface one need merely examine the direction of the outward pointing normal attached during the first phase of the construction. In the second case an analytic solution is calculated for the extremes of visibility of the cross section at the s=0 end of the cone and using the transformation

calculated earlier for s=1, the visibility limits at the other end of the cone are obtained. This determines what part of the edge of the end face pointing away from the camera is visible. The limbs (lines joining the extremes of visibility) can then be inserted. This same strategy can be used for more general sweeping rules and will need no modification for them. It should also work for other two dimensional primitives besides circular segments as long as functions are provided to find the extremes of visibility. However when the spine is no longer straight, extra complications arise and lines which are not merely swept along by the sweeping rule are introduced (see fig. 4 for an example, where a square has been swept along a circular spine and linearly halved in size along the way). This area will be investigated later.

Usually detailed information about sun angle and observer position will be available. The Observability Graph contains information which makes use of such special case information. It contains information which is not quasi-invariant. For example, sun angle and observer viewpoint information enable prediction of shadows of vertical edges; object dimensions can be inferred from single views. Much of the Observability Graph contains quasi-invariants which are deduced by the modeling program from the generalized cone representation, from the cones themselves, from their cross sections, from their limbs, and from relations between cones, between cross sections and between limbs.

i. *Cones:* Elongated cones appear as elongated ribbons from most viewpoints (over most of the solid angle of the observer hemisphere). Thus, aircraft fuselages and runways appear as elongated ribbons.

ii. *Cross Sections:* Cross sections at either end of generalized cones are typically planar. That is, cones are terminated by planes in many cases. From most viewpoints, cross sections are simply related to object cross sections. For example, circles map into ellipses and rectangles map into special quadrilaterals. Concavities are preserved. From a large range of viewpoints, circles appear nearly circular and rectangles appear nearly rectangular, because foreshortening is a cosine effect. Symmetries are nearly preserved. Alternatively, some parts are terminated by hemispheres. All projections have circular and elliptic arc segments.

iii. *Limbs:* Limbs of generalized cones are frequently straight lines. Straight lines map into straight lines. In other cases, they may be roughly circular (the limb of a donut).

iv. *Relations:* Relations between cones, cross sections, and limbs provide other quasi-invariants. Often airfields have a pair of parallel runways. Engines are parallel to the fuselage of an aircraft. Colinearity and cotermination are common relationships which are invariants. Front/behind are quasi-invariant relations. Symmetry of parts such as engines are quasi-invariant. Most parts are generated by straight spines, cylinder or cone sweeping rules, and planar termination. For them, cross sections at opposite ends of a part are related by simple plane transformations. Cross sections at opposite ends of a part are often identical, that is when a generalized cylinder is terminated by parallel planes. Limbs of generalized cylinders are parallel.

These invariants and quasi-invariants are used to map from object structures (generalized cones) to picture structures (ribbons). Each can be used to map the other way, that is from picture structures to object structures. They can be used in a descriptive way (data-driven) as well as in a goal-driven way. The system promises an interesting generality. There are more ambiguities in this direction of mapping, however techniques for resolution of ambiguities by enforcing global consistency are being developed.

The Model Matcher

The goal of the Model-Matcher is to find members of a class of objects given a generic description of that class. That is, it is designed to describe and locate any of a class of airfields, as opposed to matching a specific one. Generic object models are matched against features and relations obtained from a picture, organized into a Picture Graph. A matching process such as this faces familiar problems: in particular, errors caused by decisions made on evidence which is too local, and a combinatorial number of searches for global decisions. A familiar solution is relaxation graph matching. In our case, there is an enormous range in cost and benefit of perceptual operations. For example, runways and aircraft both indicate airfields. Runways are 50 times longer than aircraft, and have simpler shapes. Thus, they are much less expensive and more reliable to locate. Once the system finds candidates for runways, it can search for parked aircraft in a small area. The relaxation process is structured into coarse matching and detailed matching. Coarse matching uses the Observability Graph to match local properties such as shape to select initial candidates and a correspondence to the Object Graph. The next phase uses more global contextual information, as well as more detailed features to establish globally consistent matches from these screened candidates.

This scheme of coarse matching followed by detailed matching has been used in other systems. Here, a more powerful means of selection of candidate matches will be used than in previous approaches. In order to succeed in complex real-world scenes, this research seeks mechanisms for using two-dimensional shape for initial selection of candidates. It also seeks ways of using local three-dimensional interpretations of shape to limit search, by interpreting two-dimensional features as generalized cones, or cross sections or limbs of generalized cones. Garvey (1975) selected candidates by designing filters of pointwise properties such as color. Bolles (1976) used correlation of small patches to match features of a known object in approximately known position and orientation. It appears that pointwise properties are not sufficiently selective for harder problems. Similarly, correlation patches are not appropriate for generic descriptions.

Initial candidates are selected using local features and relations which have been determined to be observable by the program which utilizes the Knowledge Base. Those features and relations are organized into an Observability Graph, (OG). Both its nodes and their relations are linked to both the Appearance and Object Graphs. The nodes in the OG may be other Observability Graphs -- observables for an object may still be observable when that object is part of a larger context. Locating instances of a node is only the first part of the selection of candidates. The contextual information provided by related parts or objects of the scene will be encoded in the arcs extending from this node. Each primitive node in an OG will represent a single class of ribbons, that is, it may be viewed as a predicate which accepts any ribbon which has a certain set of attributes.

The arcs of the OG represent structural or spatial

relations expected to hold between a pair of these nodes -- examples include "intersection", "parallel", and "same-length" predicates. Assume on-1 and on-2 are nodes in an OG connected by the arc oa-1. Assume further that pn-1 and pn-2 are nodes in the Picture Graph, which will be defined soon. It is then oa-1's function to examine each pn-1 and pn-2 pair found acceptable by on-1 and on-2 to determine whether they satisfy a prescribed set of relations. For example, oa-1 may represent (that is, returns True when) an intersection at which "the instantiation of on-1" terminates while "the instantiation of on-2" does not.

In addition, it is possible to represent n-ary relations, (for arbitrary n) in an OG. An example might be "connectivity", defined as the transitive closure of intersection. To provide the scope and versatility desired, all three of these components (nodes, arcs and relations) will be implemented as general LISP functions. This format allows any component to glean information from some other part of this OG, or from any other source it wishes. Further, it allows the Knowledge Base to store only the information considered significant, sidestepping the limitations which would arise if one could only fill in a standard attribute list every time.

Throughout the following discussion, (a simplified version of) an Airport will be the canonical example of a scene. Its Object Graph can be briefly described as a collection of several runways and taxiways, close to some terminal and hanger buildings. There will probably be airplanes in the vicinity as well. The system of runways and taxiways should be connected and all these constituent parts of an airport should be in close proximity.

There are both parallel and intersection arcs between runways in the Airport Object Graph. Intersections are usually planar, not overpass intersections. Several runways may be parallel. There will usually be runways in several directions to accomodate wind changes. Further, there is often an underlying equilateral triangle pattern dating back to the time before jets, when runways were much shorter. The glide path will be free of obstructions. Runways are connected by taxiways to terminals or storage areas. A taxiway may be curved, relatively short or hard-to-see.

At the next lower level, these parts must be defined. Informally, runways must be straight, long, level, narrow and highly visible. In addition, they commonly have markings and a dotted line running down their center, and appear as roads which lead nowhere. (That is, they do not connect into the highway system.) The runway node is itself a graph. Its two nodes are both primitive. The first is the "outline" of the runway, which is a straight ribbon highly contrasting along the edges, and long. Here more specific information can used, as the range of acceptable lengths and widths are approximately known. The second ribbon is the dotted line which runs down the length of the other ribbon. The sole arc in the runway graph specifies that the dotted-line ribbon must be contained in the main ribbon and that their axes coincide.

Aircraft are described in terms of graphs whose nodes are volume parts (fuselage, wings, tail, engines) and whose primitives are generalized cones.

There are two types of nodes in the Airport Observability Graph, runways and aircraft. From almost any angle, runways appear as long, straight ribbons with constant width. They usually have markings and boundaries with high contrast. Thus their boundaries or markings are likely to be found by edge finding routines. Runways are more easily found than aircraft for this reason, as well as their length and simple shape. Thus, strategies derived from the Observability Graph are expected to focus attention on runways.

In typical examples, there will be accurate observer altitude, location and orientation and ground elevation. This will enable good approximate estimates for length and width to be made directly from the image. Under these circumstances, typical length and width are observables. In many cases, the images could be registered with familiar observables. For example, in photos of the San Francisco Bay Area, the shore can be registered, to provide a measurement scale over the whole image. Even in other situations, when these quantities could not be included in the Observability Graph, the length to width ratio could be used, as it would be large in almost any viewing situstion; and this qualifies it as an observable. In stereo viewing, measurements can be made of flatness and levelness. They would not be observables in monocular viewing. With accurate observer location and information, parallelism is accurately determined. Otherwise, in almost all cases parallelism is nearly preserved. Intersection is invariant. In stereo images, planar intersection can be determined, otherwise it can sometimes be inferred.

It is assumed that an effective edge-finding process will be combined with a spatial organization process to obtain a graph whose nodes are edges of the image and whose arcs represent spatial relations between pairs (or n-tuples) of these edges. Some particularly relevant relations between edges are: 1. colinear continuation (binary); 2. opposite, especially parallel-opposite (binary) 3. extended intersection, noting which edges terminate at this junction (binary); 4. extended coincident (n-ary star); and 5. same-length (n-ary). (See Figure 5.) There is not yet an effective edge-finding and description step such as assumed, yet there is reasonable progress in this direction with recent work by Nevatia(1977) at USC, Rosenfeld(1977) at Maryland, Barrow at SRI, as well as earlier work of Ohlander(1975), Marr(1975), and Binford-Horn(1973). Whether this process is performed uniformly or is controlled by strategies calculated from the Observability Graph will not be discussed here. Ribbons correspond to parallel-opposite and opposite relations between colinear clusters of edge fragments. These will be the primitive nodes of the "Picture Graph". Each ribbon node will also contain other information, such as internal shading and intensity contrast across its boundaries.

A Spatial Graph is constructed from stereo and 3-D cues detected in the Picture Graph. Marr calls this the 2 1/2-D sketch. Because both camera position and orientation are known, exact lengths and angles can be computed and stored in the Spatial Graph. This graph, together with the Picture Graph and the Observability Graph, will be given to the Model Matcher. From them, the Matcher will screen sub-graphs of the Picture Graph and Spatial Graph which are initial candidates for detailed matching with the Object Graph. It is essential to the coarse selection that local context be used. This means that initial selection relies not only on nodes but uses the arcs as well. Termination is a powerful cue; runways are roads that don't go anywhere. Length, width, and straightness are additional constraints. While parallelism and intersection are not required, they are unlikely as accidents; they strengthen the runway interpretation.

Each match of ON-PN subgraphs can be viewed as an interpretation of that observability subgraph in the picture. It is useful to make the interpretation by mapping from Observability Graph to Picture Graph. Typically, there will be multiple spatial relations between edges and ribbons in the Picture Graph, only some of which are consistent with the observability subgraph. It is, however, a local mapping. The goal now is to determine the best overall interpretation, one which uses the full model. Global considerations, (particularly structural or spatial relations,) will be used to determine whether a pair of ON-PN mappings is consistent -- that is, if both can be realized simultaneously. This concept is well illuminated by Escher's "Belvedre". (Escher 1967) By inspection, each of the pillars joining the upper story to the base is acceptable, when taken by itself. It is only by considering global properties, in particular how the position at the base of each support compares with its position at its top, that the architectural "flaw" can be detected. The impossibility of the total structure emerges from the fact that there is no consistent way of realizing all the pillars at the same time when the apparent relative locations of their end points is considered.

The consistency-finding algorithm now invoked regards each ON-PN correspondence as a node in the "Pairing Graph". Its first task is to use the arcs and relations of the OG to link together consistent pairs of these pairing nodes. It then removes the more isolated nodes from this graph, to leave a large and self-consistent sub-graph.

In the airfield example, the global context primarily involves distinguishing runways from portions of highways among candidate ribbons. Because there are detailed expectations for each interpretation, it is useful to consider each. Locating taxiways, storage a eas, and aircraft, nearby large flat areas, and clear flight path along alleged runways supports an airfield interpretation. On the othe hand, locating connecting highways, car traffic, buildings and obstructions along the path, supports a highway interpretation.

Thus far, the edge detection and subsequent ribbon finding process, have been simulated by hand. Also, the interfaces between the Matcher and the Knowledge Base have yet to be finalized. The matching process sketched above refers to the driver routines -- the real work will be done by the actual observability functions; that is, the arc and relation predicates, and organizing the features of a ribbon which should be used. Finally, the extensibility of this part of the system should also be noted -- new functions can be added anytime to incrementally improve the pairing evaluation process.

References

Agin, Gerald J. (1972): *Representation and Description of Curved Objects*, Stanford Artificial Intelligence Laboratory, Memo AIM-173, Oct.

Baumgart, Bruce G. (1974): *Geometric Modeling for Computer Vision*, Stanford Artificial Intelligence Laboratory, Memo AIM-249, Oct.

Binford, Thomas O. (1971): *Visual Perception by Computer*, Invited paper at IEEE Systems Science and Cybernetics Conference, Miami, Dec.

Bolles, Robert C. (1976): *Verification Vision Within a Programmable Assembly Sysytem*, Stanford Artificial Intelligence Laboratory, Memo AIM-295, Dec.

Braid, I.C. (1973): *Designing With Volumes*, Cantab Press, Cambridge, England.

Escher, M.C. (1967): *The Graphic Work of M.C.Escher*, Meredith Press, New York, Oct.

Garvey, Thomas D. (1976): *Percepual Strategies for Purposive Vision*, SRI Artificial Intelligence Center, Technical Note 117, Sept.

Horn, B.K.P. (1973): *The Binford-Horn Edge Finder*, MIT Artificial Intelligence Laboratory, A.I. Memo 285, Dec.

Marr, D. (1975): *Early Procesing of Visual Information*, MIT Artificial Intelligence Laboratory, A.I. Memo 340, Dec.

Marr, D. and H.K. Nishihara (1976): *Representation and Recognition of the Spatial Organization of Three Dimensional Shapes*, MIT Artificial Intelligence Laboratory, A.I. Memo 377, Aug.

Miyamoto, Eiichi and Thomas O. Binford (1975): *Display Generated by a Generalized Cone Representation*, Conference on Computer Graphics and Image Processing, May.

Nevatia, Ramakant (1977): Technical Report, Image Processing Institute, Sept.

Nevatia, Ramakant and Thomas O. Binford (1977): *Description and Recognition of Curved Objects*, Artificial Intelligence 8, 77-98.

Ohlander, R.B. (1975): *Analysis of Natural Scenes*, Dept. of Computer Science, Carnegie-Mellon Univ. April.

Rosenfeld, A. (1977): *Algortihms and Hardware Technology for Image Recognition*, Proceedings: Image Understanding Workshop, Minneapolis, Minn., Apr.

Sutherland, Ivan E., Robert F. Sproull, and Robert A. Schumaker (1973): *A Characterization of Ten Hidden-Surface Algorithms*, Evans and Sutherland Computer Corporation, Salt Lake City, Utah. (also published in ACM Computing Surveys, 6 (no. 1) March 1974)

Winston, Patrick H. (1975): *Learning Structural Descriptions from Examples*, in The Psychology of Computer Vision, P.H. Winston (ed.) McGraw-Hill.

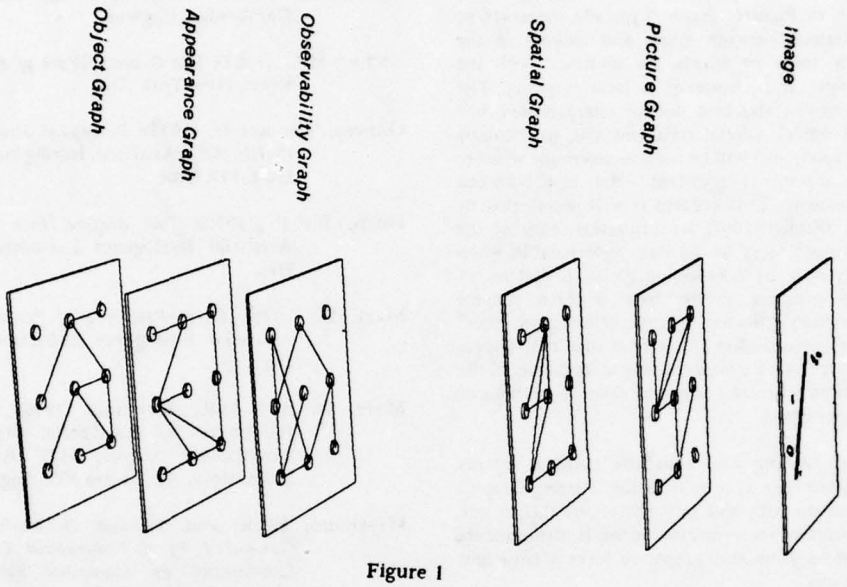


Figure 1

```
(define-object GENERIC-OIL-TANK of-class OIL-TANK
        (having-part TANK-BODY
                with rotation π/2 about y-hat
                with cross-section TANK-CROSS having
                        (type   circle
                         radius (range  (90.0 . 110.0)))
                with spine TANK-SPINE having
                        (type   straight
                         length (range  (70.0 . 100.0)))
                with sweeping-rule having
                        (type   constant)
))
```

Figure 2a

```
(define-object of-class OIL-TANK
        with position (vector 1500 1700 0)
        (having-part just-like TANK-BODY
                with cross-section just-like TANK-CROSS having
                        (radius 95.0)
                with spine just-like TANK-SPINE having
                        (length 85.0)
))
```

Figure 2b

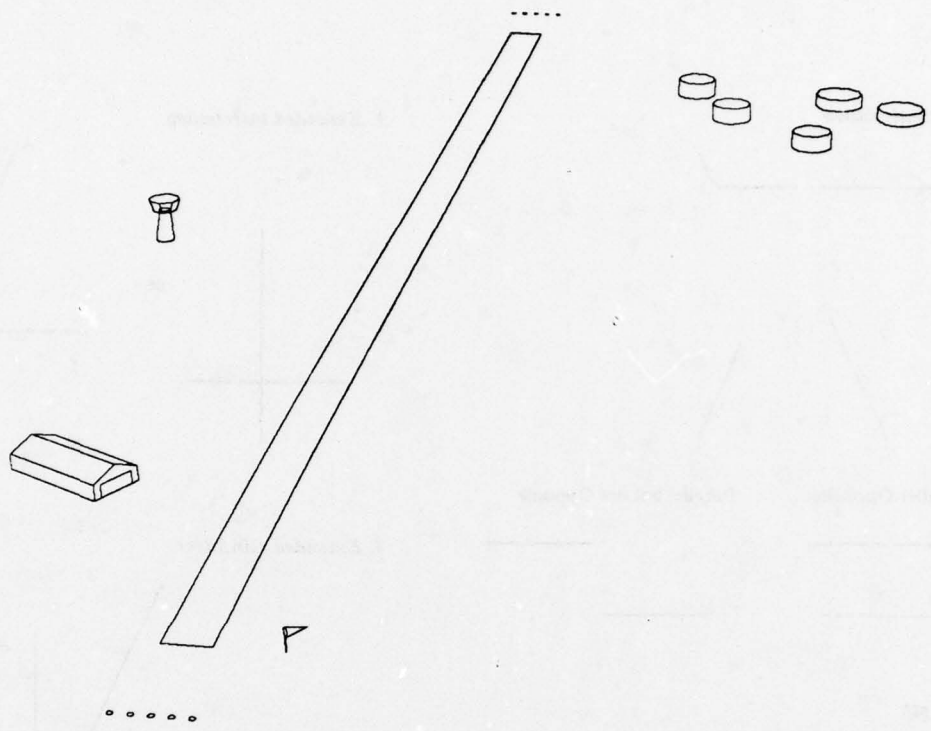


Figure 3

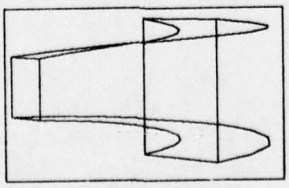


Figure 4

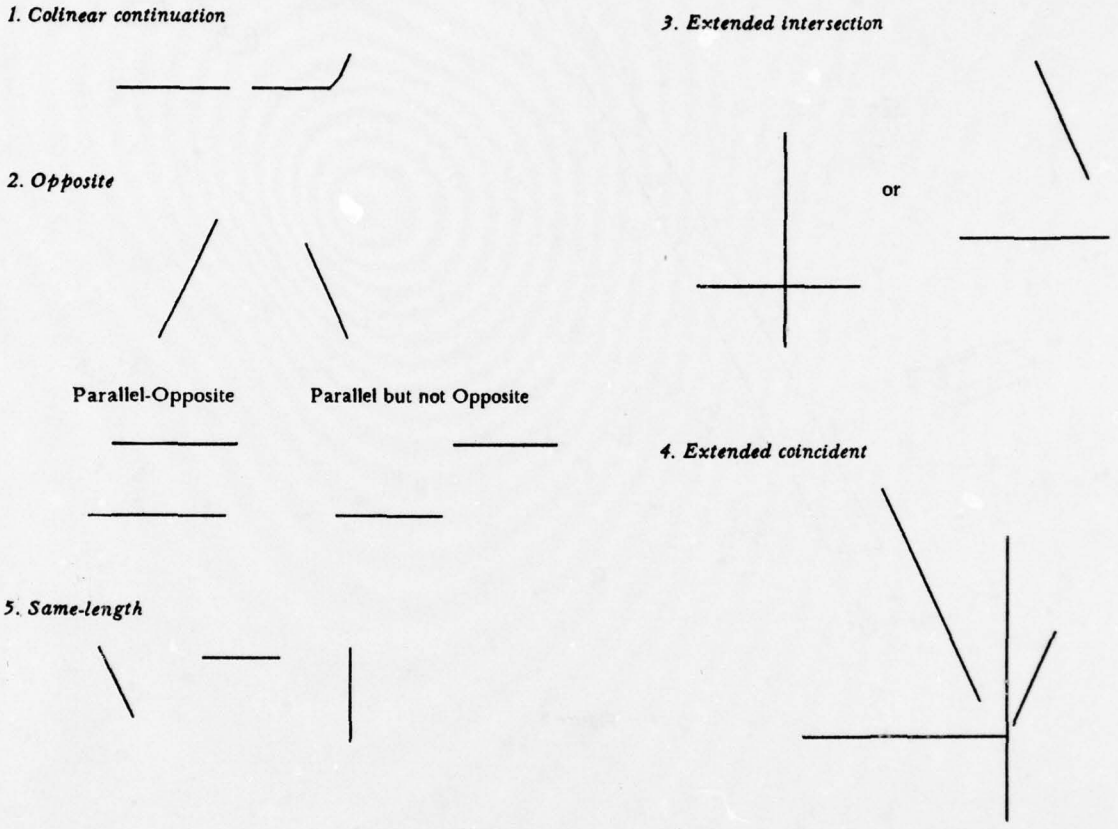

1. Colinear continuation
2. Opposite
Parallel-Opposite
Parallel but not Opposite
3. Extended intersection
or
4. Extended coincident
5. Same-length

Figure 5

Task Independent Aspects of Image Understanding

Takeo Kanade

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

Abstract

The problem of transforming picture domain cues to scene domain cues is addressed as an important task independent aspect of Image Understanding. There are several sources of task independent information, such as structural, spectral, and geometrical knowledge, that can relate the image domain cues to the scene domain cues. In this paper we present a methodology for integrated exploitation of those knowledge sources.

I. Introduction

An Image Understanding System can be roughly divided into two parts: a task dependent part and a task independent part. Although Image Understanding is characterized by an effective use of knowledge of the task domain, the performance of task independent part is in fact very critical to the final performance of the system. This paper focuses on the task independent aspects of transforming picture domain cues into scene domain cues. After discussing what kind of information (structural, spectral, and geometrical) is exploitable for this purpose, we propose to use the "Origami" world as an appropriate space for the integrated use of all the information.

II. From Picture Domain Cues to Scene Domain Cues

The term "task independent" in Image Understanding often refers to low-level image processing such as line extraction, region segmentation, etc. However, in this paper we will try to separate out the more crucial parts of the task independent aspects of Image Understanding.

It is somewhat standard in AI problem solving to employ schemes with the nature of the hypothesis-and-test paradigm; schemes which involve some "positive" feedback loop among input, cues, models, and hypotheses. One possible scheme of Image Understanding is depicted in Figure 1. Several points should be noted here. First, there is a distinction between the picture domain and the scene domain ([Clowes, 1971], [Kanade, 1977]). In short, the picture domain cues are the features observed in the picture, such as line segments, homogeneous regions, intensity gradient, etc. The scene domain cues are the features which cause the picture domain cues, such as edge configurations, surface orientation, reflectivity, lighting conditions, etc. This distinction prevents one from confusing features in the picture domain with those in the scene domain. For example, the "above" or "next-to" relationship between regions in the picture does *not* necessarily correspond to the "on" or "touching" relation between objects. The most basic and important scene domain cues are spatial three-dimensional configurations.

The second point is that general models of *concepts* are usually represented in terms of scene domain cues. See [Winston, 1970] for example. An "arch" is described by using orientation ("lying" and "standing"), spatial relation ("supported-by", "left-of", etc.) and object kinds ("brick"), all of which are terms in the scene domain.

The third point is that the right half cycle (from *image* to *model*) of our hypothesis-and-test loop in Figure 1 is more crucial to the final success of the total system. The first iteration in the loop is especially important since it provides our initial guess. Once we get a good initial guess, things work better and better. Most of the existing successful systems obtain this initial guess either by assumptions about the task environment (e.g., the boundary lines with the dark background for the Shirai's Line Finder[Shirai, 1974]), or by cooperative use of range data which simplifies the problem of obtaining scene cues (e.g., [Nitzan, Brain and Duda, 1977]).

In the context of Figure 1, the process of going from *image* to *picture domain cues* has been traditionally viewed as the task independent part of Image Understanding. In our opinion, however, the process of going from *picture domain cues* to *scene domain cues* is the more important and relevant aspect of task independent analysis. The initial hypothesis generation greatly depends on what can be done *task independently* in this process. We ought to obtain the basic *understanding* of this process before developing task specific solutions.

In the succeeding sections we will discuss what kind of knowledge, theories, and heuristics are usable for going from picture domain cues to scene domain cues, particularly for guessing the 3-D configurations of the scene, and how they can be integratedly used.

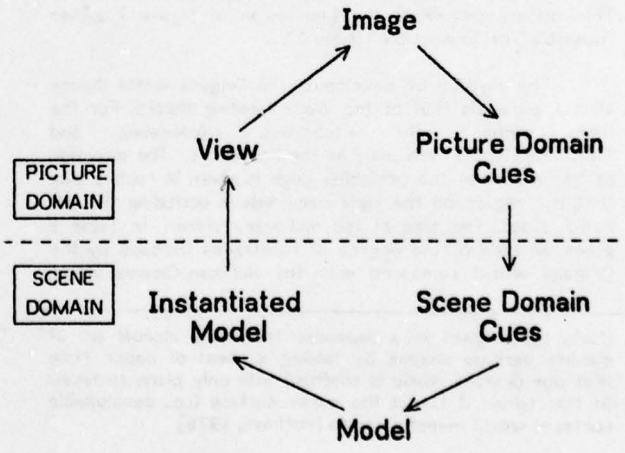


Figure 1. A Scheme of Image Understanding.

III. Structural Information

By structural information we mean the line connections and junction types of a line drawing of the scene. The Huffman-Clowes-Waltz scheme provides a method of finding the three-dimensional configurations of a line drawing of the trihedral world [Waltz, 1972]. It assigns to lines the labels which represent the 3-D meaning of the line such as + (convex), - (concave), and ← or → (occluding boundary). This method has various good features: (1) clear-cut definition of the objective world, which resulted in incorporating knowledge in a systematic way as well as eliminating vague heuristics, (2) compiled knowledge representation in the form of junction dictionary, and (3) efficient labeling procedure by filtering.

However, the scheme has serious limitations for being applied to real-world images. Besides the problems of how to accommodate missing and extra lines, the world itself is too limited. For example, the carton box of Figure 2 is an "impossible" figure. Recently I have developed a labeling

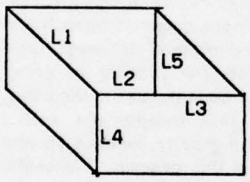


Figure 2. A Line Drawing of a Carton Box.

scheme for the world called "Origami" world [Note 1] [Kanade, 1978], which parallels Waltz's labeling scheme for the trihedral world. The key difference is that in the Origami world the *plane surfaces* themselves are the stand-alone objects, whereas in the conventional world for computer vision, such as trihedral world, the *solid objects bounded by planes* were the basic stand-alone components. This difference makes the box shape of Figure 2 either "possible" or "impossible" [Note 2].

The method of developing the Origami world theory almost parallels that of the Waltz labeling theory. For the time being, only + (convex), - (concave), and ↑ or ↓ (occluding) are used as the line labels. The direction of the arrow of the occluding edge is given in such a way that the region on the right hand side is occluding the left hand side. The size of the dictionary shown in Table 1 gives an idea of the degree of constraints imposed by the Origami world compared with the Huffman-Clowes world.

[Note 1]: Origami is a Japanese traditional manual art of making various shapes by folding a sheet of paper. Note that our Origami world is confined with only plane surfaces. In this sense it is *not* the paper surface (i.e., developable surface) world investigated in [Huffman, 1976].

[Note 2]: We can regard Figure 2 as a case where the thickness of the carton paper is not shown. It is then an *imperfect* drawing in the trihedral solid-object world. However, it is more reasonable and practical to regard it as a *perfect* drawing in the Origami world.

Junction Type	Huffman-Clowes Dictionary	Origami World Dictionary
L	6	8
ARROW	3	12
FORK	3	9
T	4	16

Table 1. Comparison of Dictionary Size between the Huffman-Clowes World and the Origami World.

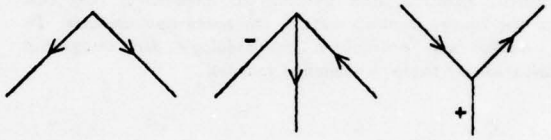


Figure 3. Some legal junctions in the Origami world.

Figure 3 shows some of the junction labels which are legal in the Origami world, but not legal in the Huffman-Clowes world. The labeling procedure is also similar except that some global check concerning surface orientation is necessary. This check can be done systematically by using the gradient space representation of surface orientation together with the compiled knowledge contained in the Origami junction dictionary (see [Kanade, 1978] for the detail). The picture of Figure 2, for example, can have 37 different interpretations in the Origami world.

The Origami world corresponds well to the way in which we would interpret a picture which has been segmented into regions. The meaning of this statement is understood by thinking why we get perfectly satisfied with the pictures like Figure 4(a) and Figure 4(b) when they are obtained as results of region segmentation of a "chair" and a "door" scene. Needless to say, the Origami world includes the solid-object world as its subset. We feel that it is rich enough to accept a much larger class of line drawings and at the same time it has enough structure to impose constraints

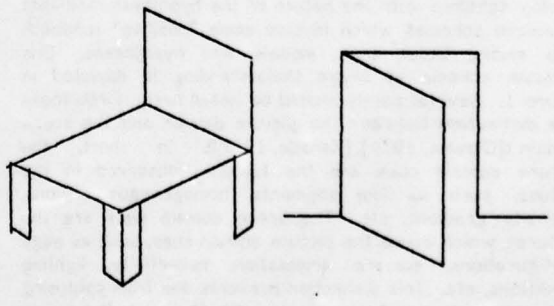


Figure 4. Region segmented pictures we think perfect: (a) chair and (b) door.

on the possible label combinations. In addition, some classes of line drawings with noise (missing or extra lines) and those of curved objects become manageable in the sense that the interpretations in the Origami world can be regarded as approximated configurations.

IV. Spectral Information

By spectral information we mean intensity and color information of image. As was shown by Horn [Horn, 1977], the image intensities carry information about three-dimensional shape; they should be used for more than just picking up line segments or segmenting a picture into regions. One convenient technique for the exploitation of this information in connection with labeling line characteristics is to examine an intensity (more generally color) profile taken across an edge. Figure 5 shows the typical types of intensity edge profiles. The use of this information can be done in two ways: absolute and relative.

The absolute method exploits the properties which give direct cues about identity of line labels. The simple rules given by Horn [Horn, 1977] are:

(Rule H-1) An edge profile with a peak shape or step with a peak superimposed suggests a convex edge.

(Rule H-2) A roof-shaped profile suggests a concave edge.

(Rule H-3) A negative peak or a step *with* a superimposed negative peak strongly suggests obscuration.

The relative method is based on the fact that if the two lines have the same edge profile, it suggests that they will likely take the same label, even though the label identity itself is not known. The classical matched T configuration is a good example. In Figure 6(a) if the edge profiles of the line L1 and L2 are similar (and preferably if the edge profiles of the lines L3 through L6 are also similar), then the labels of L1 and L2 are likely the same and the lines L3 through L6 are obscuring edges in such a way that the region R is obscuring L1 and L2. It should be noted that the geometrical information (line collinearities between L1 and L2, between L3 and L4 and between L5 and L6)) has also been used here. The matched Pi configuration of Figure 6(b) is another example which gives similar constraints. Use of color data expands possibilities of exploiting this type of constraints.

One problem with spectral information is that the constraints are often local, fragmentary, and uncertain; in some places strong evidences exist, while in others there is none. In fact, as is pointed out in [Horn, 1977], those rules (H-1) to (H-3) are not strictly necessary and sufficient conditions. Also the rule (H-3) about an obscuring edge rule

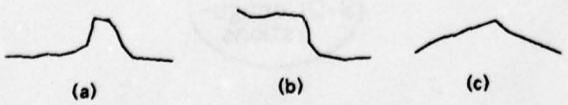


Figure 5. Typical types of intensity edge profile: (a) peak, (b) step, and (c) roof.

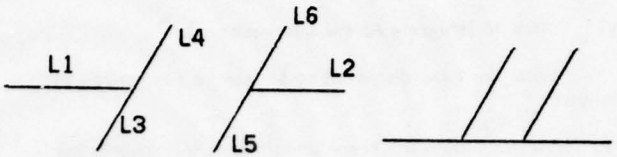


Figure 6. (a) Matched T Configuration, and (b) Matched Pi Configuration.

does not tell which side is obscuring which. In the actual image, edge profiles may or may not show clear evidence depending on various conditions. Thus we need a scheme which integrates this partial, noisy information. It is noteworthy that the relative way of using spectral information is more reliable because it is based on the comparison of edge profiles rather than the identification of particular properties.

V. Geometrical Information

By geometrical information we mean exact values of such properties as collinearity, angle, length, etc. They do not seem to tell much about scene cues directly. Rather they have to be combined with other information. The global check concerning surface orientation mentioned in section III is combined with structural information, and the matched T and matched Pi configurations in the preceding section are combined with spectral information.

Although much is not known, two things should be mentioned here. First, most of the geometrical properties in the picture domain begin to make sense only after the spatial configurations are known or hypothesized. The gradient space by Mackworth [Mackworth, 1973] is a powerful tool for relating geometry in the picture with surface orientations in the scene. Use of it together with the labeling procedure (which is essentially a gross 3-D configuration hypothesizer relying on the structural information) of the Origami world led us to an interesting algorithm for establishing relations among the "actual" orientations of the surfaces involved in the scene (see [Kanade, 1978] for the detail).

Second, it is interesting to note the following observations. Guzman's SEE program [Guzman, 1968] introduced a bunch of heuristics involving geometrical properties such as matched T and parallel background boundaries. The Waltz theory which systematically realized the Guzman's goal does not explicitly use much of geometrical information to find a unique 3-D configuration. Then why is it that computer vision researchers dealing with actual image data feel that geometrical information should be playing an important role?

One plausible explanation about these observations is the following. The Guzman's SEE used the picture domain cues and scene domain cues in a mixed way. The Waltz world, basically the trihedral solid-object world, is so constrained that it does not need to directly use most of geometrical information. However, the world in which vision researchers try to interpret real world images should be much richer than the trihedral solid-object world.

VI. How to Integrate All the Information

What we have discussed so far can be summarized as follows:

(1) How we obtain the scene domain cues (particularly the 3-D configurations) is one of the most crucial parts of an Image Understanding System. More research is required to learn how to obtain the scene domain cues in a task independent manner.

(2) The traditional trihedral solid-object world is too constrained to work with real world images. The Origami model is a good candidate for a richer world which still has a well-behaved structure.

(3) The increase of ambiguity by extending the working world to the Origami world can be offset by the integrated exploitation of spectral and geometrical information.

(4) Spectral information together with some geometrical information can be converted into constraints on the possible labels and/or label combinations for lines.

In this section we propose a method of integrating all the structural, spectral, and geometrical information in order to obtain the 3-D configurations of the scene from an image. Figure 7 shows a basic idea. An image is first segmented into regions and represented as a line drawing. Edge profile analysis is performed to obtain a set of constraints on line labels. This can be done locally for each line and as many constraints as possible should be extracted. Each constraint obtained can be represented in the form of

(<*constraint-name*> <*constraint-body*> <*confidence-value*>).

For example, when the (Rule H-1) about convex edges cited in section IV is applied to a line L, it will yield a constraint expression like

(IDENT ((L) (+)) .8)

which means that the label *IDENT*ity of the line L may be + (convex) with confidence .8. The (Rule H-3) would yield an expression like

```
(OR    (IDENT  ((L) (↑)) .9) )        [Note 3]
       (IDENT  ((L) (↓)) .9) ),
```

which means that the line L may be an occluding edge in one *OR* the other direction; i.e. though the occlusion appears to occur at the line L, it is not known which side is occluding which. As another example, the matched T configuration of Figure 6(a) will yield constraint expressions like

```
(SAME (L1 L2) .9)
(IDENT ((L3 L4 L5 L6) (↑ ↑ ↓ ↓)) .9).
```

The first expression means that the line L1 and L2 may have the *SAME* label and the second one means that a set of lines L3 through L6 will take such a combination of labels that the middle region occludes the rest.

The confidence value may be determined by the kind of rule used for deriving the constraints and the degree of matching of the edge profile characteristics. How to give the confidence value is not fully investigated yet.

Then the search process in Figure 7 takes the line drawing, the set of of constraint expressions, and the Origami junction dictionary as input. It searches for the "best" interpretations in the sense of the best constraint satisfaction in the space of the possible interpretations in the Origami world. In the present implementation, if a constraint is not satisfied a penalty as much as the confidence value of that constraint is added to that interpretation. Thus the best interpretaion means the one with the least penalty. Some additional sequential mechanisms might be needed in the future for error correction. For instance, use very confident spectral evidences first, and if a junction is an impossible one, then try to locate missed lines in the image so that the junction becomes a possible one. However, the important point is that we could convert the problem of finding the 3-D

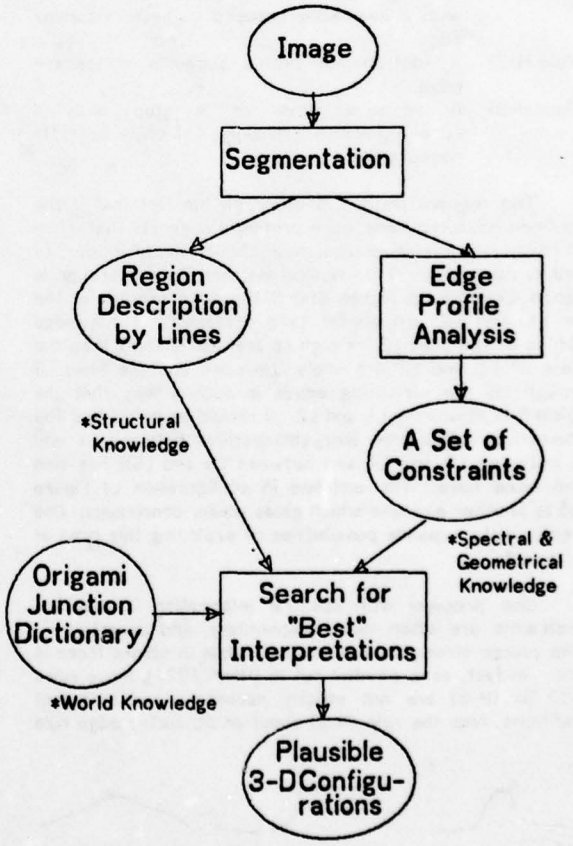


Figure 7. A Method of Integrating Structural, Spectral, and Geometrical Information to Obtain 3-D Configurations.

[Note 3]: Actually, the OR is the fuzzy logical OR operator of the constraint expressions. The fuzzy NOT is also possible. Since a set of constraints mean their conjunctions, AND is not necessary.

configurations of the scene to a problem of searching in the space of the Origami world, so that all the local, noisy, absolute, and relative evidences are exploited in a well-understood manner together with the global structure of the scene.

As a simple example, let us take a case that an image of a carton box is given, and let us explain how the proposed method would work for guessing the box shape from that image. The region segmentation process produces (hopefully) a segmentation which is represented as a line drawing of Figure 2. The edge profile analysis would yield the following set of constraint expressions:. (The confidence values are given provisionally here, because they are not the central issue at this point.)

```
(OR    (IDENT ((L1) (↑)) .9)
       (IDENT ((L1) (↓)) .9) )
(OR    (IDENT ((L2) (↑)) .9)
       (IDENT ((L2) (↓)) .9) )
(OR    (IDENT ((L3) (↑)) .9)
       (IDENT ((L3) (↓)) .9) )
```

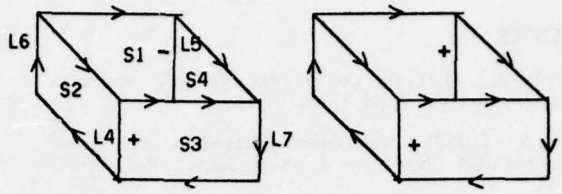
! These are obtained by the rule (Rule H-3) applied to L1, L2, and L3.

```
(OR    (IDENT ((L4) (+)) .7)
       (IDENT ((L4) (-)) .6) )
(OR    (IDENT ((L5) (+)) .7)
       (IDENT ((L5) (-)) .7) )
```

! These correspond to the case where the edge profiles of L4 and L5 are found not to have a negative peak property but it is not clear whether they are a peak shape or roof shape.

If we search for the interpretations which satisfies these constraints among the possible interpretations in the Origami world, then the configurations of Figure 8 are obtained as the first two most plausible ones. Figure 8(a) is in fact the box shape configuration we wanted. Note that the above constraints obtained by the spectral information alone did not tell the directions of the obscuration at L1, L2, and L3, nor the definite identities of the line characteristics of L4 and L5, and that the line drawing of Figure 2 alone can have 37 different configurations. However, if they are integratedly used the desired 3-D configuration (box shape) of the object is discovered as one of the most plausible ones.

Some people might question about the significance of these results of labeling to the total Image Understanding process. First, they tell which surfaces are related to which surfaces. For example, a simple algorithm can show that the labeling of Figure 8(a) means that the surface orientations of S1, S2, S3, and S4 should have the relations in the gradient space as shown in Figure 9(a). The gradient space is in short a parameter space of plane surfaces and a point in it represents the orientation of the plane relative to the viewer. If we assume that the lines L4, L5, L6 and L7 are parallel in the picture, the gradients (surface orientations) of S1 through S4 should be on a line and the ordering relations between the surfaces connected by arcs should exist as shown. Therefore the effect of partially knowing or hypothesizing about the surface orientations has been explicitly represented in the diagram. For example, a hypothesis that the surface S1 and S3 are parallel (i.e. gradient points of these two overlap) results in the ordering between S2 and S4 as shown in Figure 9(b).



(a) (b)

Figure 8. Some Plausible 3-D Configurations of Figure 1. (See text about the given constraints.)

Second, the labelings are the cues to access the models of concepts. Consider the "simplest" (thought not so) task about the image of the above example; to know that the object in the scene is generally called a "box". In order to know that the object can be named a "box", it must be known, at least partially, that the image can have the box shape, which has been done in Figure 8(a). In fact, this is the point emphasized in section II by saying that the process of going from image domain cues to scene domain cues is the important task independent aspect.

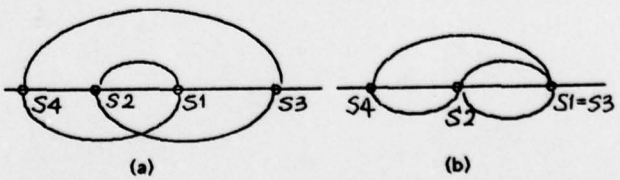
VII. Conclusion

This paper presents a methodology of how the structural, spectral, and geometrical information can be integrated to obtain the 3-D configurations of the scene from the image. One major claim is that the Origami world provides useful constraints in integrating the above information from real images, because it accepts a large class of line drawings and still has enough structure. In fact it corresponds the manner in which we interpret the region segmented picture.

The complete theory of Origami World is presented elsewhere [Kanade, 1978]. The search program in Figure 7 with a simple set of constraints is working. This is a report of work in progress, and we plan to apply the proposed method to real images as well as investigating various kinds of constraints extractable from images.

Acknowledgement

I would like to thank Raj Reddy for his valuable suggestions and comments for writing this paper, and Dave McKeown for his critical reading of the manuscript.



(a) (b)

Figure 9. Gradient space representations of the relations among surface orientations: (a) Relations corresponding to Figure 8(a) (b) Relations after hypothesizing that S1 and S3 are parallel.

REFERENCES

Clowes, M. B. (1971). "On Seeing Things", *Artificial Intelligence* Vol. 2, No. 1, pp.79-116.

Guzman, A. (1968). "Computer Recognition of Three Dimensional Objects in a Visual Scene", *MAC-TR-59*, MIT.

Kanade, T. (1977). "Model Representations and Control Structures in Image Understanding", *Proc. IJCAI-77*, Vol. 2, pp.1074-1082.

Kanade, T. (1978). "Origami Understanding", Technical Report, Department of Computer Science, Carnegie-Mellon University (in preparation).

Horn, B. K. P. (1977). "Understanding Image Intensity", *Artificial Intelligence*, Vol. 8, No. 2, pp.201-231.

Huffman, D. A. (1976). "Curvature and Creases: A Primer on Paper", *IEEE Trans. Computer*, Vol. C-25, No. 10, pp.1010-1019.

Mackworth, A. K. (1973) "Interpreting Pictures of Polyhedral Scenes", *Artificial Intelligence*, Vol. 4, NO. 2, pp.121-137.

Nitzan, D., Brain, A. E., and Duda, R. O. (1977). "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis", *Proc. IEEE*, Vol.65, No.2.

Shirai, Y. (1973). "A Context Sensitive Line Finder for Recognition of Polyhedra", *Artificial Intelligence*, Vol. 4, No. 2, pp.95-119.

Waltz, D. (1972) "Generating Semantic Descriptions from Drawings of Scenes with Shadows", *MAC AI-TR-271*, MIT., reproduced in *The Psychology of Computer Vision*, (Winston, P. ed.), McGraw Hill, 1975.

Winston, P. (1970) "Learning Structural Descriptions from Examples", *MAC AI-TR-76*, MIT., reproduced in *The Psychology of Computer Vision*, (Winston, P. ed.), McGraw Hill, 1975.

ROAD TRACKING AND ANOMALY DETECTION IN AERIAL IMAGERY

Lynn H. Quam
SRI International
Menlo Park, California

ABSTRACT

This report describes a new procedure for tracking road segments and finding potential vehicles in imagery of approximately 1 to 3 feet per pixel ground resolution. This work is part of a larger effort by SRI International to construct an image understanding system for monitoring roads in aerial imagery.

INTRODUCTION

This report describes a new procedure for tracking road segments and finding potential vehicles in imagery of approximately 1 to 3 feet per pixel ground resolution. This research is part of a larger effort by SRI International to build a "knowledge based road expert," described by Barrow and Fischler elsewhere in these proceedings.

The overall effort is directed towards specific problems that arise in processing aerial photographs for such military applications as cartography, intelligence, weapon guidance, and targeting. A key concept is the use of a generalized digital map data base to aid in the interpretation of imagery.

OBJECTIVES

The primary objectives of the overall "road expert system" are to analyze images to:

(a) Find road fragments in low- to medium-resolution images

(b) Track roads in medium- to high-resolution images

(c) Find anomalies on roads

(d) Interpret anomalies as vehicles, shadows, signposts, surface markings, etc.

The road tracking algorithm discussed here is started with the position of the center and direction of a road fragment found by part a). The nominal road width is supplied either from the data base or by an image analysis function that can determine the width of a road fragment. The road tracker produces two forms of output: a point list describing the track of the road center and a binary image of all points in the road that are anomalous and might belong to vehicles. In the complete road-expert system, this image will then be analyzed by part d) to screen false alarms and interpret the remaining anomalies.

ALGORITHM DESCRIPTION

Figure 1a shows a representative road scene containing segments of a multilane freeway, with a few vehicles and road surface markings (painted arrows and words in the leftmost lane). The wear patterns in the lanes correspond linearly with the road. The vehicles and other anomalies stand out as being quite different from the pattern of the road.

The basic road-tracking algorithm exploits the above observations. Successive road intensity cross-sections (RCS) taken perpendicular to the direction of the road showed a high degree of correlation, which suggested that road tracking could be accomplished by using cross-correlation. The location of the correlation peak was used to maintain alignment with the road center and to generate a model for the road trajectory. However, this approach turned out to be unsatisfactory because small alignment errors accumulated and anomalies perturbed the correlation peak.

To overcome these problems, four refinements were introduced:

(a) Cumulative road cross-section model

(b) Trajectory extrapolation

(c) Anomaly detection

(d) Masked correlation.

Instead of aligning consecutive RCSs, each RCS is aligned with a cumulative RCS model, based on an exponentially weighted history of previously aligned RCSs. Parabolic extrapolation of past correlation peaks is used to predict the future road trajectory. The predicted trajectory is used to guide the tracker past areas where the correlation peak is unsatisfactory. Anomalies are detected by comparing the aligned RCS with the RCS model. Corresponding pixels that significantly disagree are marked as potential anomalies. The cross-correlation is then repeated, masking out the anomalous pixels to obtain a more accurate alignment.

Steps for the refined tracking algorithm are given below:

(1) Based on past road center points and directions, extrapolate the position of the road center K feet ahead.

(2) Extract the road cross section (RCS) intensities along a line perpendicular to the direction of the road at the extrapolated center point.

(3) Use cross-correlation to find displacement of the current RCS with respect to a model (RCS model) that is dynamically constructed by the road tracker.

(4) Generate a mask indicating the positions of anomalous pixels that deviate from the RCS model.

(5) Recorrelate over the unmasked pixels.

(6) Update the RCS model using only the valid points of the current RCS at the best alignment. Update is done using an exponentially decaying average.

(7) Adjust the position of the road center according to the location of the correlation peak.

(8) Detect anomalies as being significant deviations from the RCS model.

(9) Repeat steps 1-8 until the edge of the image is encountered or the RCS model becomes invalid.

EXPERIMENTAL RESULTS

In the experiments shown here, the road tracker was interactively started by indicating the following information for each road segment:

<X0,Y0>	center of road lane
theta0	direction of road at <X0,Y0>
w	nominal width of road

The freeway example in Figure 1 conforms well to the above model, as shown by the overlay results in Figure 1b. The bright lines indicate the road trajectory and the bright blobs indicate potential anomalies.

The simplistic model that a road consists of well-correlated intensity cross-sections clearly breaks down in the example shown in Figure 2a, where the road surface changes from concrete to asphalt on the overpass. Certainly the RCS model generated for the asphalt will not match the intensities in this globally changed road surface.

When the tracker encounters the surface change a high percentage of the pixels in the RCS will be anomalous (Figure 2b). When this occurs, the tracker extrapolates ahead and tries to reacquire the road. If the road is not reacquired within the length of the longest expected anomaly, the tracker then assumes that a pavement transition has occurred and establishes a new RCS model.

Most of the anomalies marked in Figure 2b are due to road surface changes. All four vehicles were found also. A later section will discuss basic changes to the control structure of the current program to eliminate the false alarms occurring from the surface changes.

Figures 3a and 3b show results for a freeway interchange on-ramp loop. This example is interesting since the road curves rather tightly, and the road surface changes at approximately the same place where the road trajectory changes from a circular arc to a straight line.

Figures 4a and 4b show a very complicated example of road forks, changes in lane width, and intersections. For the lanes tracked, all vehicles and at least portions of the road surface marks (arrows and words) were found. In a developed road expert system, the data base should help significantly in handling the complexities of this image by knowing the locations of forks, intersections, lane-width changes, etc. This information will help in interpreting the cause of RCS model changes.

In marked contrast with the situation in most of the previous figures, figure 5a shows a rather poorly defined dirt road with little evidence of wear patterns. Figure 5b shows the successful results of the road tracker. Most of the anomalies marked were due to shadows cast by sparsely foliated trees.

DISCUSSION

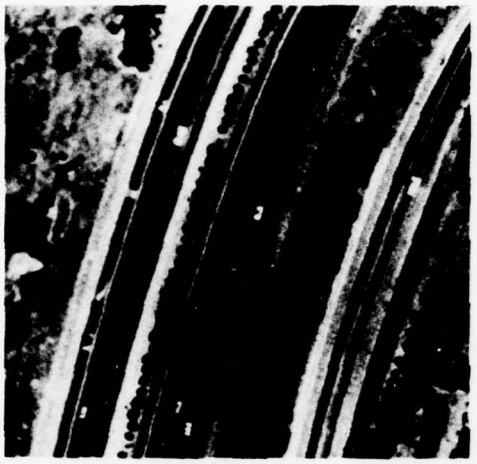
The preceding examples demonstrate the capabilities and limitations of the present tracking algorithm. The algorithm has shown surprising ability to contend with a wide variety of road situations, including total change in the road surface. The use of masked cross-correlation techniques eliminates the potential perturbances to the road track by anomalies. Trajectory extrapolation enables the tracker to reacquire the road after detecting that the road surface has changed. All results were obtained using the same program and the same detection and threshold criteria; no attempt was made to "fine-tune" the parameters individually for each example.

One defect of the present algorithm is the attempt to do too much in one pass along the road. In particular, in the present system, anomaly marking begins before road-surface changes have been detected. The false alarms created by this defect can be eliminated either by backtracking when a road transition is found, or by performing the detailed anomaly detection as a second pass along the road, using the road-course and surface-change knowledge produced by the tracker.

The road tracker presently operates as an independent module. As a component of a larger road-expert system, it will be started from the output of a map-guided road-detection algorithm operating on lower-resolution imagery. Data-base knowledge can also be used in the tracking algorithm to increase reliability and reduce false alarms in anomaly detection. Such knowledge might consist of previous imagery of the same area or geometric knowledge about locations of road forks, intersections, overpasses, surface changes, lane-width changes, etc. To exploit such knowledge, it is necessary to establish geometric correspondence between the image and the data base coordinate system. If, for example, a road anomaly corresponds to a known surface marking on the map or appears in the same place in previous images, then it is probably a surface marking rather than a vehicle. Similarly, the use of an illumination model can help to distinguish objects casting shadows from surface markings.

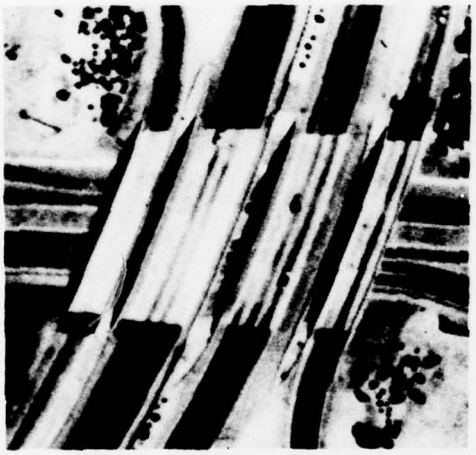
We plan to acquire and digitize images taken under diverse viewing conditions such as partial cloud coverage, snow cover, oblique viewing angles, and seasonal variations. This will introduce a new set of problems for the tracking algorithm such as non visibility of road segments due to clouds or occluding objects and major photometric differences between images of the same area. The use of a map data base and sources of knowledge will be essential to guide the interpretation of such images.

With the planned enhancements and improvements, it should be possible to detect potential vehicles with very high hit rates and low false alarm rates in difficult imagery. This capability is a central component of an overall road-monitoring system.

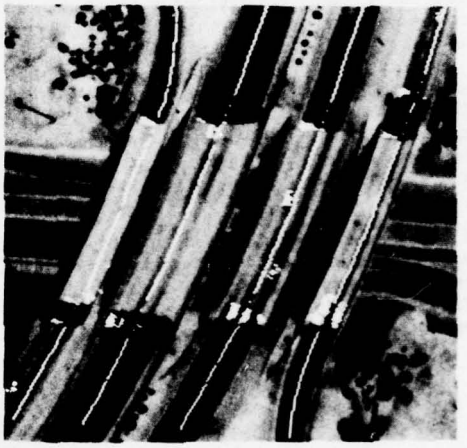


1a

1b

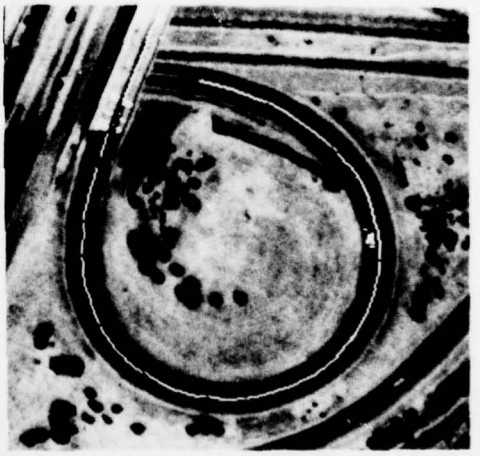


2a



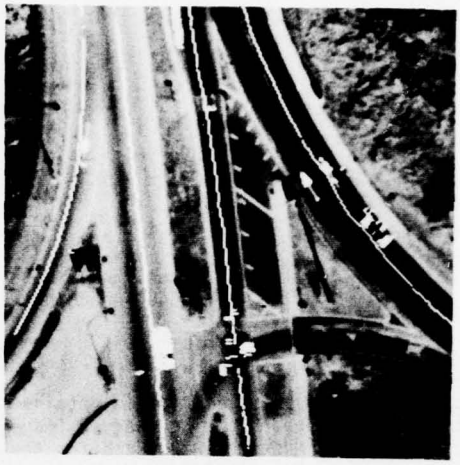
2b

3a

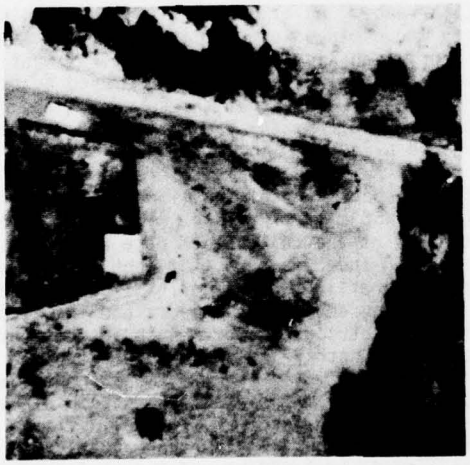


3b

4a



4b



5a

5b

DESTRIPING SATELLITE IMAGES

B. K. P. Horn and R. J. Woodham

Artificial Intelligence Laboratory, Massachusetts Institute of Technology,
545 Technology Square, Cambridge, Massachusetts 02139

Abstract. Before satellite images obtained with multiple image sensors can be used in image analysis, corrections must be introduced for the differences in transfer functions of these sensors. Methods are here presented for obtaining the required information directly from the statistics of the sensor outputs. The assumption is made that the probability distribution of the scene radiance seen by each image sensor is the same. Successful destriping of LANDSAT images is demonstrated.

1. Destriping of images obtained using multiple sensors.

An image sensing device using a single photoelectric sensor which is mechanically scanned across the scene produces outstanding digitized images since sensitivity, resolution and transfer functions are the same for all points in the image. Unfortunately, such a device is limited in speed by the mechanical movement. More importantly, it is limited in speed by the fact that an accurate measurement of scene radiance requires the collection of an adequate number of photons. This explains the preponderance of linear arrays of sensors and area sensors such as vidicons which are otherwise deficient because of geometric distortions, non-uniform response, non-uniform resolution, and so on.

A compromise can be struck, where a small set of sensors is mechanically scanned to collect the image. In the system used aboard LANDSAT, for example, each spectral band is scanned using six sensors at the same time. Thus, six lines of the image are produced during a single sweep of the mirror. On the next sweep, the satellite has advanced its orbit by an amount which allows the same set of sensors to pick up the next six lines of the image.

Unfortunately, the sensors do not have identical transfer functions. As a result, images produced in this fashion show undesirable, regular "striping". This effect can be removed if the transfer functions are accurately known, since one could then compute scene radiance from the sensor output using the inverses of these transfer functions. The sensors used in the older equipment in particular have time-varying behavior. Photomultipliers, for example, show a drift in both gain and offset (dark current) due to small changes in the material of the dynodes used in the electron multiplier stages and temperature variations.

If a reference object containing all scene radiances of interest were in the scene, one could recalibrate the sensors continuously. This is difficult to arrange. An alternative is the scanning of a gray wedge placed over a light source at the end of every scan line. This, in fact, is what is done aboard LANDSAT. The results are used to estimate the gains and offsets of the sensors. The digital data produced from the raw satellite signals is corrected using this information.

Unfortunately, one finds that the striping effect is not removed in this fashion; the reasons for this are not entirely clear. One cause appears to be the use of the calibration data as a means of adjusting gain and offset so that each sensor is related to its preflight condition. Slight changes in the light source, the gray wedge and the geometry of imaging introduce drifts which are not compensated for. Another reason is related to the fact that photomultipliers are somewhat nonlinear and have a response which depends on their exposure history. Modern devices using solid state photodiodes do not suffer from these problems.

The methods explored here for destriping images are based on the assumption that each sensor is exposed to scene radiances with approximately the same probability distribution. The sensor values can then be modified so that each one is related in the same way to the actual scene radiance. The information required to perform this modification is extracted from statistics of the observed sensor outputs.

2. A simple method for linear transducers.

If the image sensors are linear and time invariant, a simple method can be used to reduce striping. The sensor output, x', can be written as a function of the scene radiance, x, as follows:

$$x' = f(x) = a + b \cdot x$$

Each sensor has its own, fixed values of offset, a, and gain, b. If these are known, the scene radiance can be calculated using the inverse of the transfer function,

$$x = g(x') = (x' - a)/b$$

If this is done for each sensor in turn, striping effects will be removed.

The required constants for each sensor can be determined if a calibration object containing two or more known scene radiance values is available in the scanned scene. If such a calibration object is not available one can estimate the (relative) values of gain and offset using simple statistics of observed sensor values. Each sensor sees a subimage consisting of every nth line (when n sensors are used). The complete image is formed by interlacing these subimages. It seems reasonable to suppose that, for a large enough image, each subimage has approximately the same probability distribution of scene radiance values. One would not expect a particular subimage to contain many more values in a particular range of scene radiance than another subimage.

If this assumption is correct, then the gain and offset constants can be estimated from the mean and standard deviation of the measured sensor output values. If the mean of the scene radiance is μ and the standard deviation is σ, then the mean of the sensor output will be $\mu' = a + b\,\mu$ and the standard deviation of the sensor output $\sigma' = b\,\sigma$. Then,

$$b = \sigma'/\sigma$$

and

$$a = (\mu'\,\sigma - \mu\,\sigma')/\sigma$$

Clearly, it is not reasonable to assume that one can find the absolute values of the mean and standard deviation of the actual scene radiance. Fortunately, for destriping purposes only relative values are important. That is, one can use the mean and standard deviation of the sensor outputs for the whole image in place of the mean and standard deviation of the scene radiance. Naturally, now the results will not be scene radiance values. The striping however will be removed since each subimage now has the same mean and standard deviation, and, if the assumption introduced earlier applies, the same linear relationship to scene radiance.

Note that one can relax the assumption about the relationship of the subimages. Here it is not necessary that they have the same probability distribution of scene radiance, only that their means and standard deviations be the same.

3. Shortcomings of the simple method.

We have found this method to be only partially successful in destriping LANDSAT images. One reason for this may be that out of a range of 128 possible sensor outputs a range of only around 30 values correspond to normal scene radiance values. Low values are not found in short wavelength bands because of light scatter in the air. Conversely, large values correspond to cloud, snow and ice, and scene radiance values of such areas often exceed the highest available sensor output values and so result in clipping. This nonlinear effect will skew the calculation of means and standard deviations. (Low value clipping also occurs because of NASA's destriping and possible other reasons.)

One may eliminate such areas by removing sensor values above a certain level from consideration. Slightly better results are obtained in this fashion. Naturally the arbitrarily selected threshold will tend to introduce inaccuracies of its own. One way around this problem is to eliminate the same fraction of high values from the output of each sensor. The fraction to be removed can be estimated by guessing the fraction of the image which is covered with cloud, snow or ice. This is certainly better than using a fixed threshold directly on the sensor outputs.

Even with this refinement, results are not entirely satisfactory. Superficially, it appears that different gains and offsets are appropriate for different scene radiance ranges. That is, the sensor transfer curves are somewhat nonlinear. We thus devised a method which deals with this problem directly.

4. Preliminary considerations.

Consider a random variable X with probability density function $p(x)$. The function $p(x)$ is non-negative and satisfies

$$\int_{-\infty}^{\infty} p(x)\,dx = 1$$

The probability density function $p(x)$ can be estimated from a large number N of observations of the random variable X. If n of these measurements fall in the interval $[x, x + \delta x]$, then n/N tends to $p(x)\,\delta x$ as N becomes very large and δx small (in a fashion which allows $N\,\delta x$, and thus n, to become large also).

The cumulative probability density function $P(x)$ is defined as

$$P(x) = \int_{-\infty}^{x} p(t)\,dt$$

This function is monotonically non-decreasing since $p(x)$ is non-negative. $P(x)$ represents the probability that the random variable X takes on a value less than or equal to x.

Now consider observing the random variable X by means of a transducer with transfer function $f(x)$. Its output can be thought of as a new random variable X', say, with a probability density function $p'(x')$. This function is related to the probability density function $p(x)$ of the original random variable X, in a fashion which depends on the transfer function $x' = f(x)$. It is easiest to develop this relationship in terms of the cumulative distribution functions $P(x)$ and $P'(x')$ where

$$P'(x') = \int_{-\infty}^{x'} p'(t)\,dt$$

If x' lies in a range R' when x lies in the range R, then clearly,

$$\int_{R'} p'(x')\,dx' = \int_{R} p(x)\,dx$$

Now assume that f(x) is monotonically non-decreasing. Then the range $x \leq x_0$ is mapped into the range $x' \leq f(x_0)$

$$P'[f(x)] = P(x)$$

As a result one can determine the transfer function f(x) if the cumulative probability density functions P(x) and P'(x) are known and if the latter has an inverse. Then,

$$f(x) = (P')^{-1}\,P(x)$$

If P' is monotonically increasing, the required inverse will exist. Difficulties will be encountered only when P'(x) is constant over a certain range. That is, if $P'(x') = c$ [and hence $p'(x') = 0$] for $x' \in [x_1', x_2']$. Then, if $P(x) = c$, one can say only that $f(x) \in [x_1', x_2']$.

There are two possible causes of this problem. First it may be that f(x) actually has a discontinuity. In this case, one correctly finds a jump from x_1' to x_2' in the solution. The other possibility is more serious. If $p'(x') = 0$ because $p(x) = 0$ [where $x' = f(x)$ as before], then the transfer function f(x) cannot be found in the specified range because in essence no inputs are available to test it in this range. The information to recover f(x) there is thus not available.

Note, however, that if the inputs to the transducer are in fact characterized by the given probability density function, then our lack of knowledge of the transfer function in the specified range is of no consequence since there are no inputs falling in this range anyway.

To calculate scene radiance from sensor values, we actually need the inverse g(x') of the transfer function. This can be found just as easily. If,

$$P'(x') = P\,[g(x)]$$

Then,

$$g(x') = p^{-1}\,p'(x')$$

The same considerations regarding the existence of the inverse p^{-1} apply here as those discussed regarding the existence of the inverse $(P')^{-1}$. All these special case problems are avoided if the cumulative probability distribution functions are monotonically increasing.

The method shown here for finding the transfer function of a transducer (or its inverse) is based on the same analysis as that used to design a generator of pseudo-random numbers with a desired probability distribution function p'(x) when a generator is available which produces pseudo-random numbers with known probability distribution function p(x).

5. Transducer with discrete output values.

Essentially the same method may be used if the transducer produces discrete outputs. Consider, for example, a case where the input range can be broken up into a number of intervals such that

$$f(x) = i \quad \text{if } x \in [x_i, x_{i+1})$$

The probability density function of the output of the transducer is then discrete and,

$$p_i' = \lim_{\varepsilon \to 0} \int_{x_i}^{x_{i+1} - \varepsilon} p(x)\,dx$$

Clearly, $p_i' \geq 0$ and

$$\sum_{i=-\infty}^{\infty} p_i' = 1$$

The cumulative probability density function can be defined as follows,

$$P_i' = \sum_{a=-\infty}^{i} p_a'$$

If f(x) is monotonically non-decreasing, then the same argument applied in the continuous case, leads again to

$$P'\,[f(x)] = P(x)$$

If P' can be inverted, the transfer function can be found using

$$f(x) = (P')^{-1}\,P(x)$$

The only difference is that here f(x) is a function from a continuous range to a discrete domain. Naturally, when one finds the inverse of the transfer function, g(x'), using these methods, one has to accept the fact that the actual value of x cannot be recovered, only a range $[x_i, x_{i+1})$.

6. Estimation from a finite number of samples.

To apply this method to determine the transfer function of a real transducer, the cumulative probability density functions must be determined from a model of the underlying process generating the random variables or estimated from frequencies of observed occurrence using a finite number of samples. In the latter case an uncertainty (i.e., sample deviation) will be found in the estimation of the probabilities which will be inversely proportional to the square root of the number of samples falling in a particular interval.

Clearly, then, the transfer function can be estimated with limited accuracy. Accuracy will be least for ranges which happen to contain fewest samples. Thus the largest errors in determining f(x) will tend to occur where p(x) is small. In fact, as we have seen before when p(x) becomes zero over a range of values of x, then f(x) cannot be determined uniquely for this range.

The largest errors in pinning down g(x') will occur where p'(x') is small.

7. Application to satellite images.

To use this method of determining arbitrary monotonic non-decreasing sensor transfer functions to satellite images obtained using multiple sensors, one has to make the assumption that the subimages have similar statistical properties. This seems reasonable, at least if the whole image is large enough. One also has to assume that the sensor transfer functions are constant at least for the time taken to scan one scene.

The probability distribution function of the actual scene radiance is not available and so only relative adjustments can be made. That is, instead of this function, one uses the probability distribution function of the sensor outputs for the whole image as a reference. The result will be that each processed subimage has the same probability distribution function. If the assumption that all sensors are exposed to the same distribution of scene radiance holds, then this implies that the same monotonic non-decreasing functional relationship holds between scene radiances and image values. That is, striping will have been removed.

8. Details of the algorithms.

The first step is the determination of a cumulative histogram of sensor values for the whole image as a reference. Let there be H(x) occurrences of sensor outputs less than or equal to x out of a total of N values. Now for the subimage produced by sensor i, one calculates a similar cumulative histogram. Let $H_i(x')$ be the number of sensor outputs less than or equal to x', produced by sensor i, out of a total of N_i values. Here,

$$N = \sum_{i=1}^{n} N_i$$

where n is the number of sensors.

A lookup table g(x') is now constructed by applying the inverse of the function H(x) to $H_i(x')$. This lookup table is then used to modify all the sensor values produced by sensor i. The inverse can be calculated relatively easily since H(x) is a monotonically non-decreasing function. The lookup table value g(x') is the smallest number x such that

$$N_i \, H(x) \geq N \, H_i(x')$$

This process is repeated for each sensor in turn, until all image values have been modified by the lookup table appropriate to the sensor with which they were measured.

9. Results and Conclusion.

This method has been applied to part of a LANDSAT image extracted from CCT (Computer Compatible Tape). Some of the bands showed rather heavy striping. In figure 1, for example, is shown Band 6 (.7μ to .8μ in the near infrared). Applying the method described here considerably reduces the regular striping. The overall effect is that each subimage is related in the same way to the underlying scene radiance. At the same time, the overall distributions of tones is not disturbed. The result is shown in figure 2. Some localized striping is still apparent, but the regular pattern has been removed.

It is instructive to inspect the inverse transfer function for each sensor. These are shown as six subfigures of figure 3. The short horizontal sections in the transfer function correspond to sensor values which do not occur because of a particular data compression algorithm used on LANDSAT. It will be apparent from inspection of these inverse transfer functions that the sensors are somewhat nonlinear. This explains why the simple destriping technique described earlier fails.

One channel (band 7) on LANDSAT is equipped with silicon photodiodes instead of photomultipliers. Striping is apparent in data of this band (.8μ to 1.1μ) as well, as shown in figure 4, and can be removed by the technique presented here as shown in figure 5. The differences in transfer function in this case however appear to be simple gain differences as shown by the six subfigures of figure 6. So for this band the simple destriping method which assumes linear transfer functions works equally well.

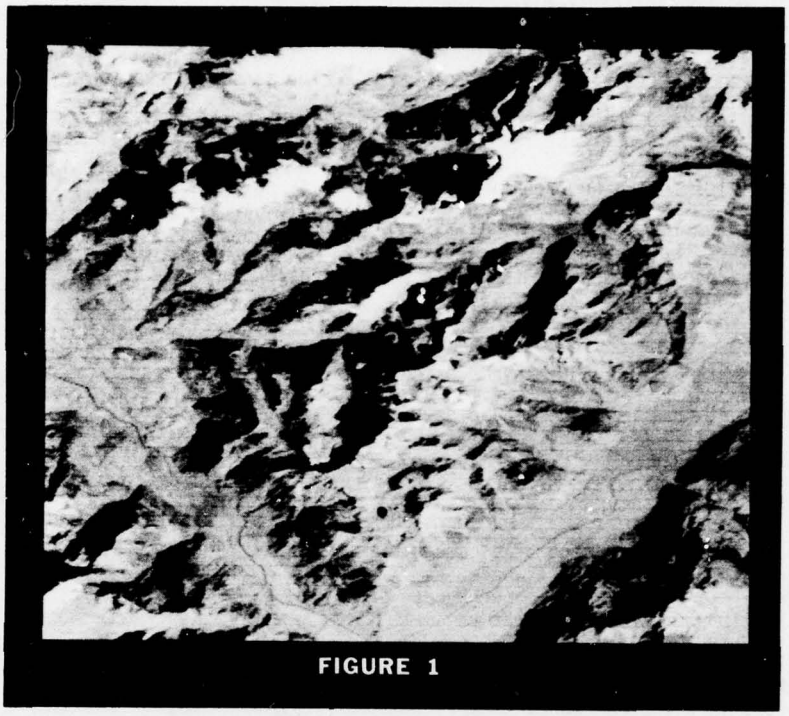


FIGURE 1

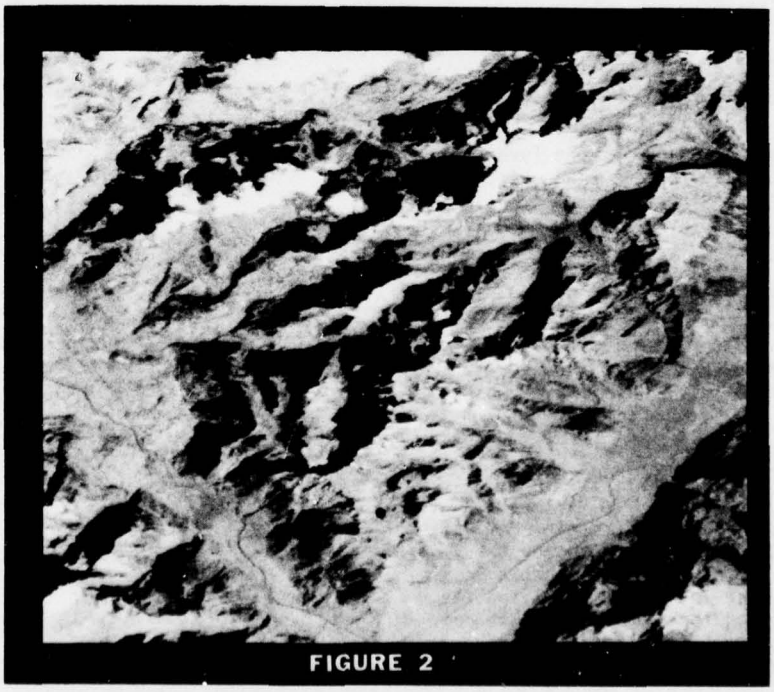


FIGURE 2

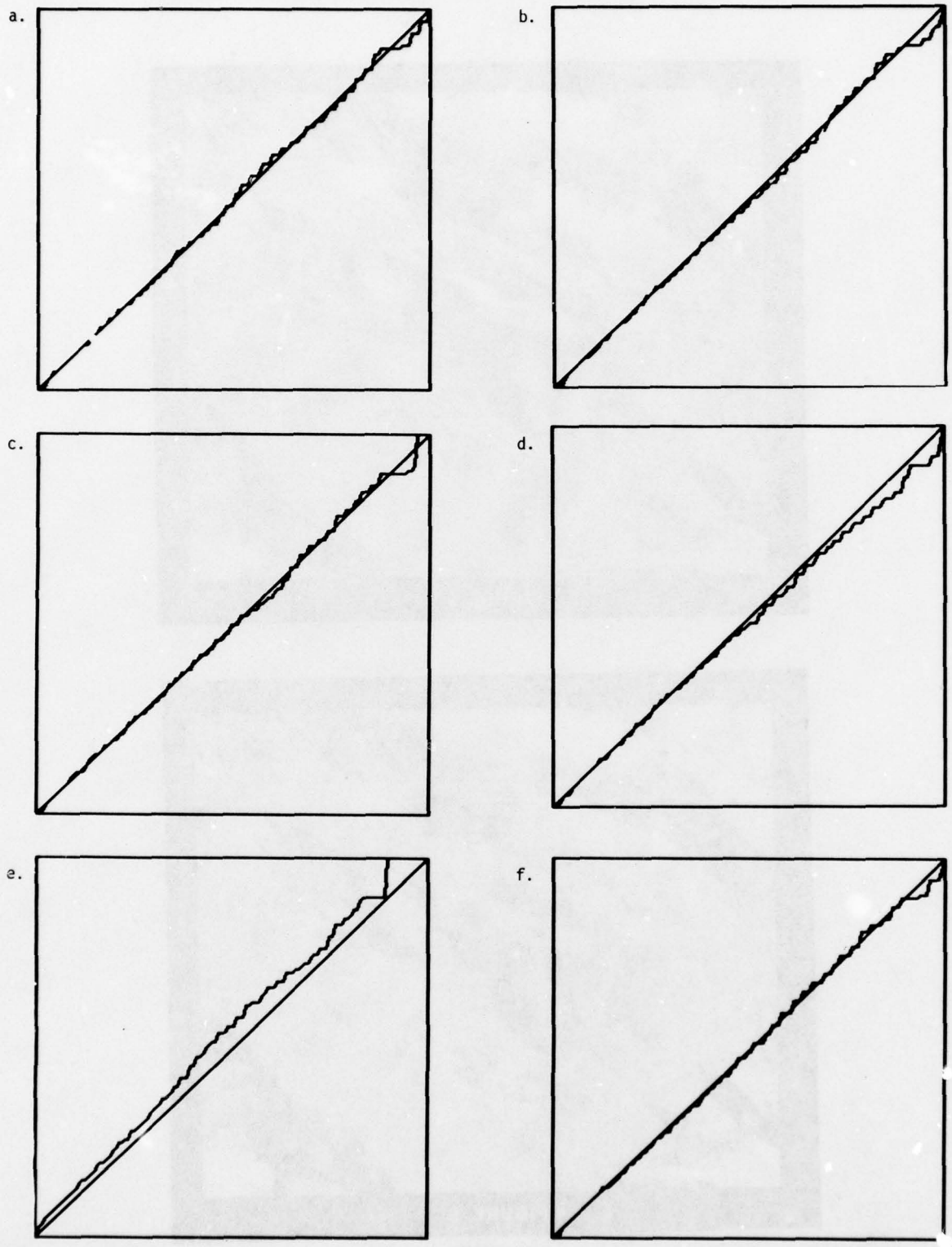


FIGURE 3

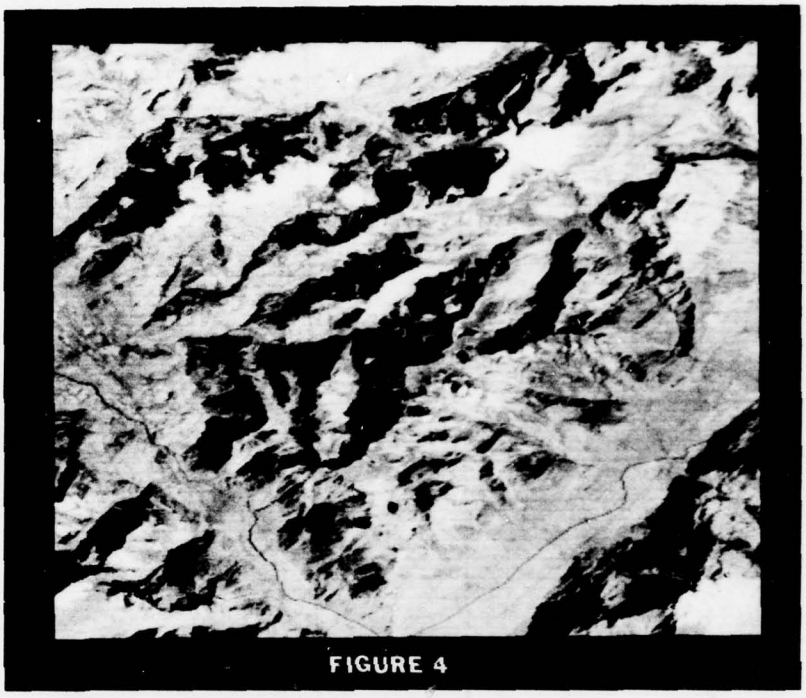


FIGURE 4

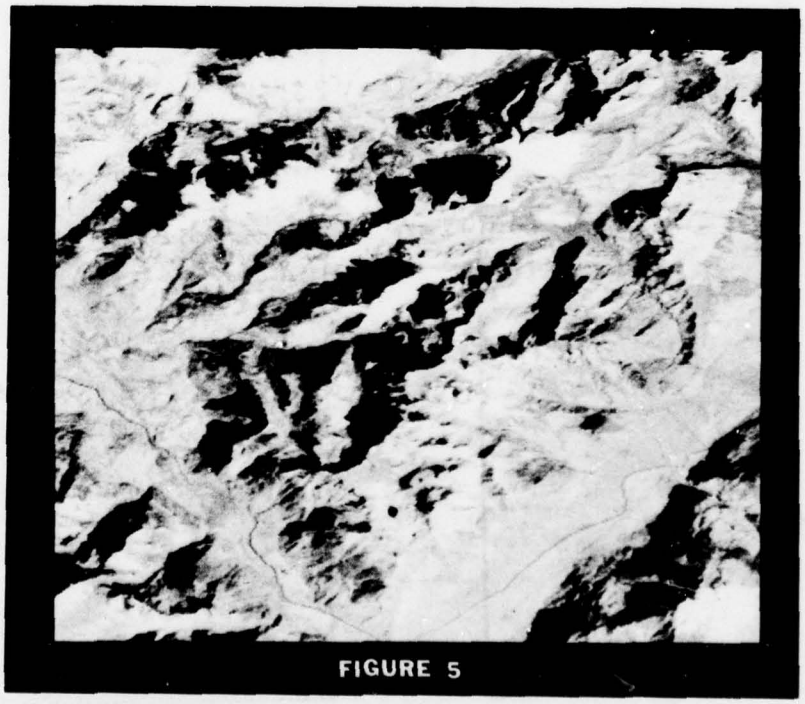


FIGURE 5

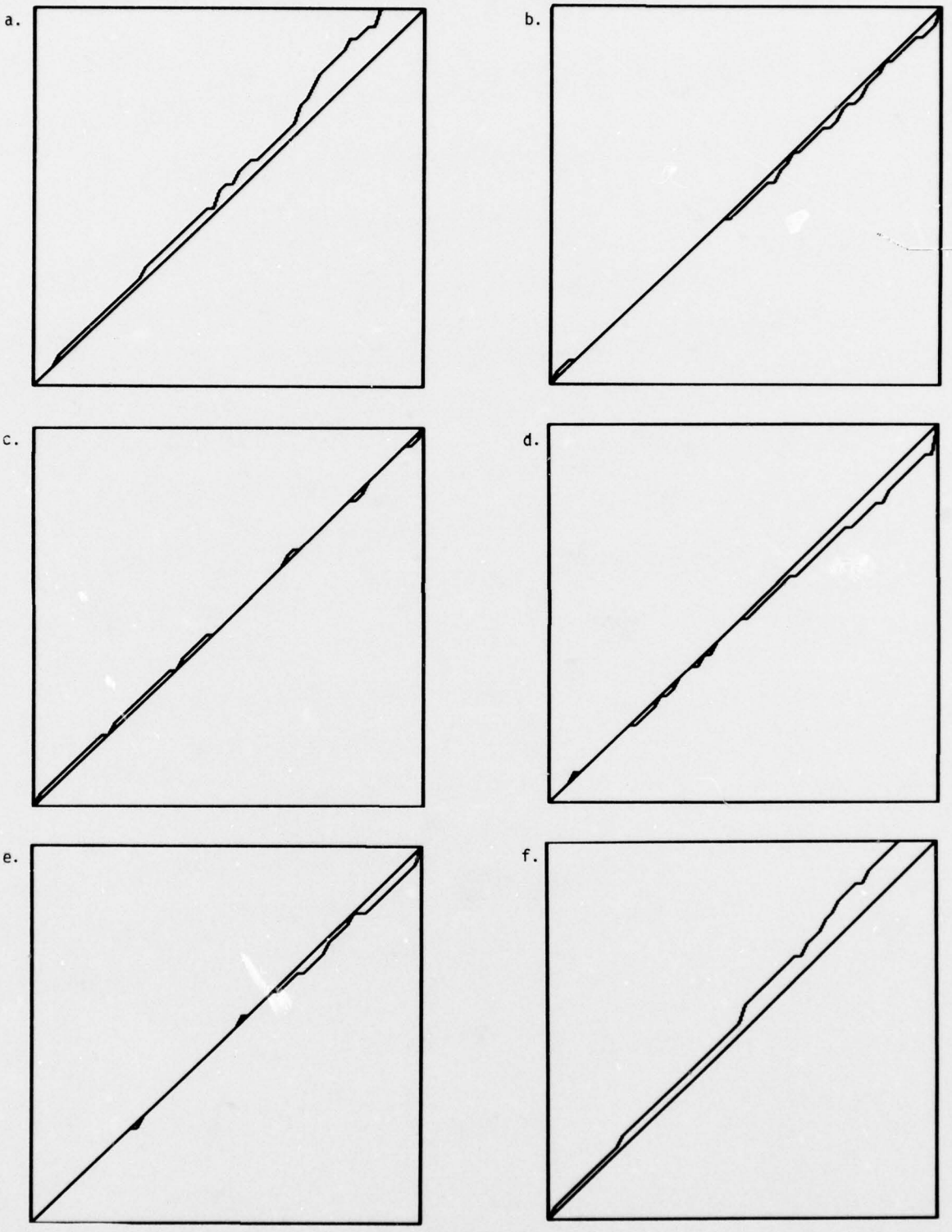


FIGURE 6

SESSION III

TECHNIQUES I

LOCAL CONTEXT IN MATCHING EDGES FOR STEREO VISION

R. David Arnold

Computer Science Department
Stanford University
Stanford, California 94305

ABSTRACT

This paper describes a stereo vision system based on edge matching. Depth maps of edges have been obtained with sequences of aerial photographs of aircraft, buildings and cars, allowing accurate measurement of heights, dimensions, and angles of surfaces of objects. The edge-based approach enables accurate determination of boundaries of objects, is effective with thin objects such as poles, and offers advantages in speed. Total computation time was 90 seconds with 128 x 128 images, with no effort at optimization.

INTRODUCTION

Stereo vision has long been important in photo interpretation and mapping and has potential applications in guidance. This research seeks mechanisms to automate stereo vision to interpret stereo images and provide three dimensional measurements of objects from those images. The objective is to demonstrate these capabilities in potentially practical implementations. The method should be fast, accurate, make full use of the resolution of images, and be able to handle a wide variety of data from either stereo or motion parallax.

The system matches corresponding features in pairs of images, rather than matching small corresponding areas by cross correlation. The features are edge elements (edgels) produced by the Hueckel edge operator. This approach offers advantages in speed and accuracy and avoids some fundamental problems of area correlation.

In an edge-based system, computation effort can be concentrated on the edges and depth information about planar surfaces inferred from boundaries. If high speed, specialized processors are used for edge operators [1], overall computation can be cut significantly. The proportions and size measurements of the boundaries are also useful for subsequent identification.

Typically, edge-based techniques offer a factor of 10 improvement in accuracy over correlation methods. In correlation, accuracy near a boundary is limited to a fraction of the width of the correlation window (typically 8x8). The Hueckel edge operator, however, provides measurements to a fraction of a pixel, even for weak or noisy edges. Edge-based systems also have an advantage with small objects. Poles and other long, thin objects are prominent features, but are too small for correlation windows.

A serious deficiency of area correlation is failure at surface discontinuities. Simple area correlation techniques inherently fail in the vicinity of surface discontinuities because the edge of an object appears against a different background area in each view of the stereo pair. It is important to locate surface discontinuities, since it is precisely the boundaries of objects where accurate measurements are most important. However, edge operators are ineffective in the presence of texture and smooth shading. In those cases, edge-based techniques encounter problems, while correlation is effective. Thus, edge-based and area-correlation approaches are complementary.

In any stereo system, ambiguity is a major problem. Edges in one view may match with multiple edges in the other view. For example, in the parking lot scenes, edges of cars, pavement markings and shadow edges are all parallel and are easily confused. There are few techniques that can reduce such ambiguity when matching individual edgels. Direction, brightness and contrast measurements extracted by the edge operator can guide the matching , but are not strong conditions.

However, ambiguity can often be eliminated by considering a local context which is larger than a single edgel. If a scene edge has continuity in three dimensions, then we expect adjacent, matching edgels along that edge to be continuous in both direction and disparity. Furthermore, intensities and colors on one or the other side of the edge should be consistent. Edge continuity and consistency are strong conditions that significantly affect ambiguity.

The context of the ground surface is also important in this matching process. Techniques of Moravec [2] and Gennery [3] are used for automatic determination of the camera model parameters and ground surface equation directly from the pictures. The knowledge of the camera model imposes the strong limitation of matching features in only one dimension. A priori constraints may be used during matching to limit the disparity range to that of objects above the ground within a reasonable height.

IMPLEMENTATION

The data are 512 x 512, 8 bit image pairs digitized from a small (3 cm square) region on each of two 9x9 inch black and white aerial photograph negatives. Subjects include commercial aircraft at a terminal in San Francisco airport (see Fig. 1), cars in a parking lot, and an apartment building complex, To date, most work has been on 128x128 images, either averaged 4:1 or selected as a window from the larger pictures. This has allowed smaller memory requirements and simpler debugging, but memory management has been implemented to allow the

techniques to work on much larger images. Execution times given below refer to the KL-10 processor at the Stanford Artificial Intelligence Laboratory.

A camera model and ground plane are calculated from the data in the images in a process which is entirely automated. An Interest Operator [2] is applied to the left view to select approximately 50 "interesting" points. A point is "interesting" if it promises to be easily locatable in two dimensions (ie. corners and intersections). A fast binary search correlator [2] produces an initial match for each point, searching the entire right image each time. These matches are refined with a high resolution area correlator [3] and passed to a camera model solver [3]. This camera model program solves for four parameters:

1) direction of the stereo axis
2) relative rotation between left and right views
3) scale factor between left and right views
4) translation perpendicular to the stereo axis

(The usual camera solver determines 5 parameters. This form is useful in the degenerate case in which scene heights are small with respect to distance from the film plane.) The relative positions (disparities) of each matched pair along the stereo axis provide information on heights relative to the film plane. At this stage, about half the original 50 points have been automatically rejected for various reasons, and we rely on the remainder to be evenly distributed in the scene. The points and their heights are given to a ground plane finder [3] which attempts to fit a plane to a subset of them such that few points are assigned heights below the plane, some may be above the plane, and as many as possible lie on the plane. Total processing for camera model and ground plane is about 8 seconds.(See Fig. 2.)

The next step is to raster-scan an edge operator over the two pictures to extract all usable edges. We use the Hueckel operator, with an operator radius of 3.19 (32 pixels area). The Hueckel operator produces several accurate measurements which can be useful in discriminating matches, including a measurement of angle that is more accurate than other operators. Information retained for each edgel includes x-y position, angle of edge, and brightness of minus and plus sides. About 1200 edgels are produced from a 128x128 picture in about 18 seconds. (See Fig. 3.) At this point, all information is contained in the edge files, and the original images are set aside. The edges from the left and right pictures are then adjusted with the camera model and ground plane parameters, to a standard coordinate system with the stereo axis in the x direction and disparity shifts due to the tilt of the ground plane cancelled. Thus all points lying on the ground plane will have identical x-y coordinates in the two views.

We now proceed to match edges in the left (master) with those in the right, and extract a local context for each edge in the left. A grid of 8x8 cells is set up for the left and right pictures, each cell being the head of a linked list. Edge records are read in and linked to an appropriate cell based on the x-y coordinates of the edgel. For these pictures, the linked lists have an average length of about 4. For each edgel in the left picture, we want to find a list of possible matching edgels in the right picture. The search is constrained to those edgels within a narrow band, about 6 pixels wide in the y direction. The band starts at the x coordinate of the left edgel (zero disparity) and extends to the a priori disparity limit in the x direction. The differences in brightness and angle are thresholded to determine whether to accept or reject a potential match. If the match is accepted, a disparity is calculated by projecting the right edgel to the y coordinate of the left edgel. On the average, this search produces 8 ambiguous matches for each edgel, that is, 8 edgels that agree in position, angle and brightness. Most of these ambiguous matches are actually multiple edgels from the same scene edge, with slight deviations in disparity due to noise. From this point on, no further information is obtained from the right edge file.

For local context, we want a list of edgels in the left picture that probably lie on the same physical edge of the object. Again, a scan runs through all edgels on the left, and a search is made for each one, this time in the left grid. Two edgels are linked if certain loose conditions are met:

1) x and y coordinates match within 3 pixels
2) their angles match within 90 degrees
3) the angle of a line connecting edgel centers lies between the individual edgel angles
4) brightnesses are consistent on at least one side of the edgels

Typically, this produces 3 or 4 links per edgel, and linked edgels tend to follow edges of low to moderate curvature. (See Fig. 5.) Time for the matching and linking is 33 seconds.

We now have for each edgel in the left picture a list of possible disparities and a list of neighboring edgels which are linked to it. The problem is to choose a disparity for each edgel in such a way that disparities are consistent along linked edges. We have implemented an ad hoc "voting" scheme whereby each disparity on the edgel's list is a candidate, and only those neighbors which are linked can vote. (See Fig. 6.) Let E be an edgel and L an edgel linked to E. Let dL be a disparity on L's disparity list and dE a disparity on E's disparity list. If dL and dE are equal or nearly equal (within .125 pixel disparity) then dE gets two votes. If dL and dE are close (within .375 pixel disparity) then dE gets 1 vote. Otherwise, there are no votes. This loose condition for voting compensates for quantization error in the recording of disparities and allows multiple edgels from a single edge to reinforce. After all the voting, a bell-shaped distribution usually results about the best disparity, with wild or inconsistent matches out on the tails of the curve. The maximum of the distribution is taken as the disparity for E. This processing takes 8 seconds. We can now output a file of edgels with their three dimensional locations.

PROBLEMS

The method outlined above suffers from some serious problems. It relies heavily on the edge operator. While the Hueckel may be one of the best choices available, it is deficient in several respects. First, it is susceptible to slow gradients, finding a multitude of parallel edges that tend to match at every possible disparity (see Fig. 4). Second, it is a least squares process, and so is easily led astray by a few bad points. For example, the direction returned for the edge becomes very innaccurate as soon as a corner enters the operator window. Finally, strong texture confuses most edge operators and could prevent the operation of this system in many regions. Assuming we can detect these conditions and avoid false matches, we are still left with many places where boundaries will have gaps that must be filled by other techniques.

Ambiguity is a fundamental problem that must be solved by any stereo system. While many ambiguities can be resolved within a given context, there will always remain some that require still wider contexts. For example, a checkerboard presents an ambiguity problem that cannot be solved until the context includes the boundaries of the pattern. We are in the process of designing several improvements, including a more extensive surface context, and a relaxation network to extend the use of context in a controlled way. It is possible to consider the voting mechanism of the current implementation as the first iteration of such a relaxation, with each successive iteration extending the context and reducing the remaining ambiguity.

The type of scene is a crucial factor in evaluating the performance of a stereo technique. In general, this system will work well in scenes of man-made objects and poorly in natural scenes. For area correlation, the situation is just the opposite. The reason is that man-made objects (cars, buildings) tend to have planar surfaces of uniform intensity and well defined linear edges. Natural surfaces (clouds, trees, hills), on the other hand, are often curved with strong texture and indistinct or irregular boundaries. A general purpose vision system would need to employ both types of techniques, perhaps even within a single scene.

RESULTS

Results are illustrated in the photographs below (see Figs. 7-10). The technique seems fairly successful, and there is strong reason to believe that with the additional context now being designed very effective stereo modeling will result.

ACKNOWLEDGEMENT

I wish to thank Tom Binford for his many contributions and guidance in this research.

This research was performed at Stanford Artificial Intelligence Laboratory under ARPA contract MDA 903-76-C-0206.

REFERENCES

1. Nudd, G. R., P. A. Nygaard, and J. L. Erickson, "Image-Processing Techniques Using Charge-Transfer Devices," Image Understanding Workshop, Palo Alto, Ca., October 1977.

2. Moravec, H. P., "Towards Automatic Visual Obstacle Avoidance," Fifth IJCAI, MIT, August 1977.

3. Gennery, D.B., "A Stereo Vision System for an Autonomous Vehicle," Fifth International Joint Conference on Artificial Intelligence, MIT, August 1977.

Figure 1. A 128x128x8 bit image pair. The scene is San Francisco Airport and the aircraft is an L-1011.

Stereo axis:	3.71 degrees
Relative rotation:	-1.06 degrees
Scale factor:	.980
Translation:	8.41 pixels

Ground plane: $z = 6.80 - .00925x - .0125y$

Figure 2. Camera model and ground plane parameters for the aircraft images.

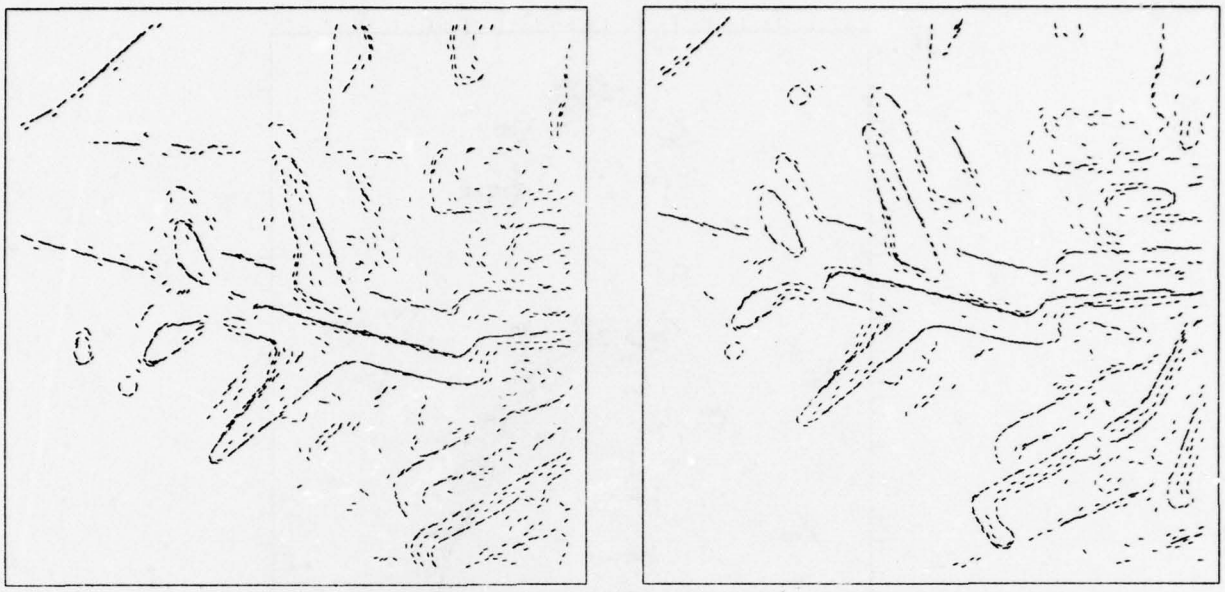


Figure 3. Results of the Hueckel edge operator. There are approximately 1200 edgels from each view.

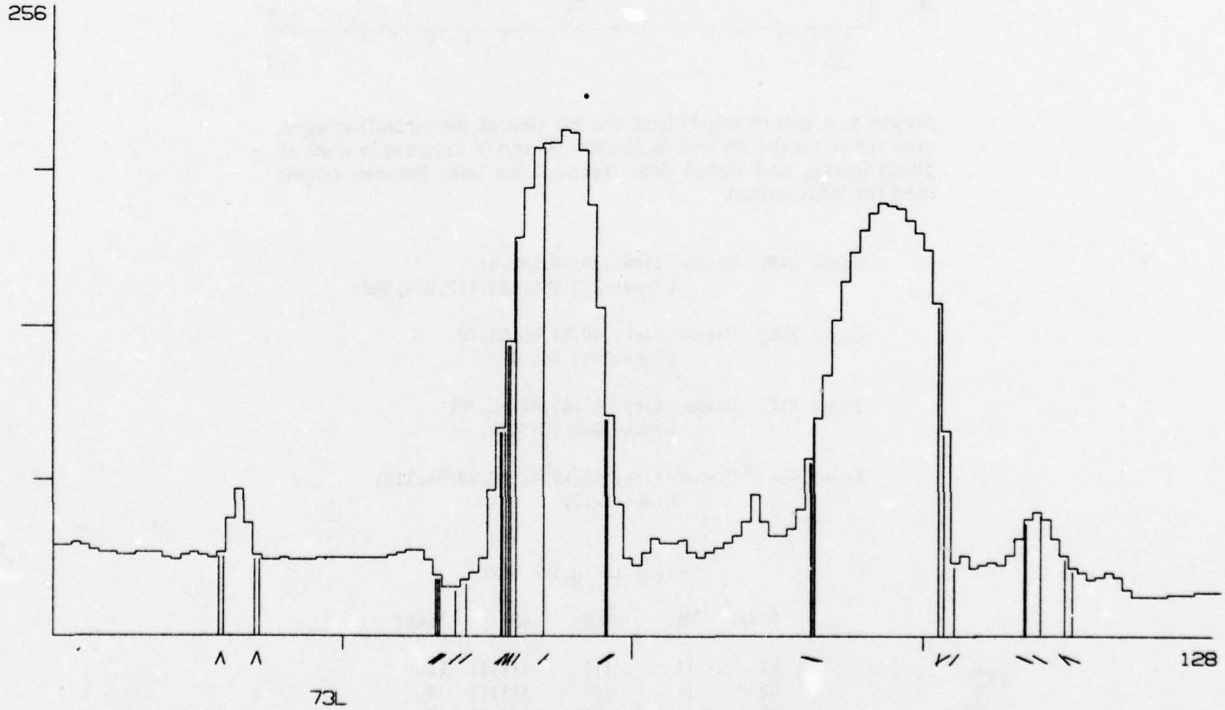


Figure 4. An intensity profile from the left view of the aircraft. The cut is taken along the stereo axis at y coordinate 73. Edgels that intersect the cut are plotted as vertical lines, with their direction indicated by the small line segments below. The cut is taken through the right wing, just grazing the fuselage. Note the multiple edgels on several of the edges.

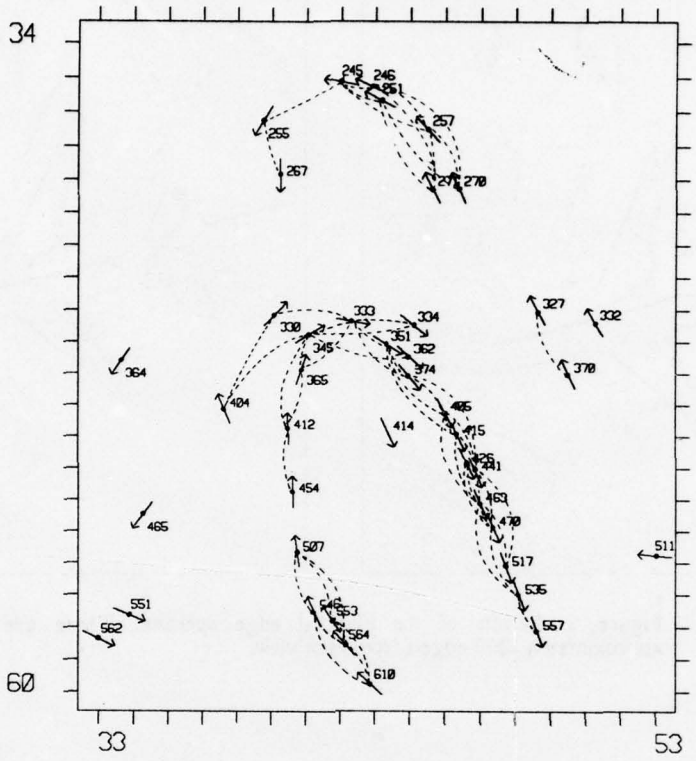


Figure 5. A plot of edgels from the left view of the aircraft images, near the left stabilizer and its shadow. X and Y axes are in units of pixels (octal), and dotted lines represent the links between edgels used for local context.

```
Edge: 345;  Disparities: 34,40,54,60;
                  Links: 333,365,404,412,334,362;

Edge: 365;  Disparities: 40,44,46,65,76;
                  Links: 333,345,412;

Edge: 412;  Disparities: 41,41,42,45,75;
                  Links: 345,365,454;

Edge: 454;  Disparities: 42,42,42,46,50,64,112;
                  Links: 412;


                  Voting tally for 412:

     Disp.   345     365     454     Total

     41      ||      |||     ||||||  11*
     42      |       ||      ||||||   9
     45              ||||    ||||||  10
     75              ||               2
```

Figure 6. A portion of the data structure produced by the matching program, and a sample voting. The edgels are selected from those in figure 5. (All numbers are in octal.)

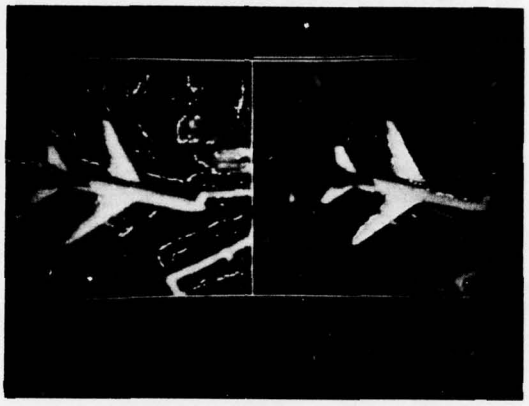


Figure 7. Results of the stereo system on the aircraft images. Edgels are shown superimposed on the video of the left image. The plot on the left shows all edgels whose disparities were determined to lie between -1 and 1 (pixels) (edgels on the ground surface). On the right is a plot of edgels between 2 and 3.5 pixels disparity (main wings).

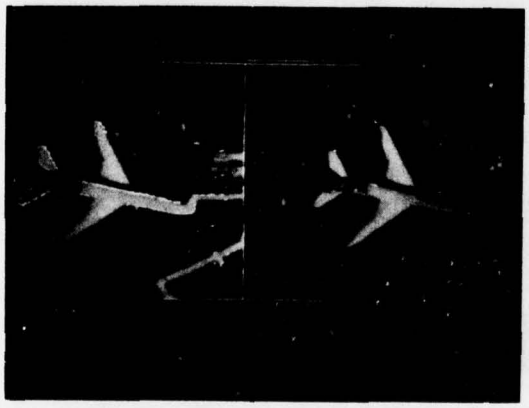


Figure 8. Edgels between 3.5 and 6 are plotted on the left (fuselage and stabilizers), and between 6 and 9 on the right (boarding ramps).

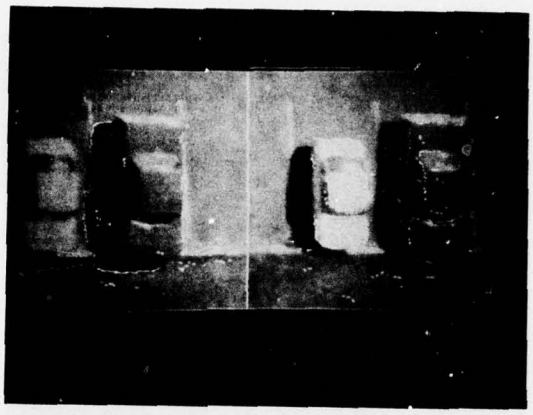


Figure 9. Results with parking lot scene. Disparity ranges of -1 to 1 and 3 to 12 include edges on the ground and on the cars, respectively.

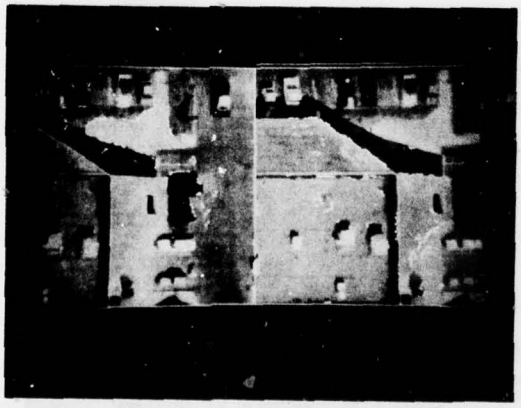


Figure 10. Results with building scene. Disparity ranges of -1 to 2 and 6 to 17 separate the ground from the roof.

THE CORRESPONDENCE PROCESS IN MOTION PERCEPTION

Shimon Ullman

The Artificial Intelligence Laboratory
Massachusetts Institute of Technology

1. The correspondence problem

The visual perception of motion requires the establishment of *correspondence* between elements in the scene [Ullman 1977a]. To be seen in motion, a moving element has to be perceived first at one location, then at another. The two images of the element, at the two locations, have to be identified as representing the same element in motion, and this identification is termed the correspondence process. In this paper the correspondence problem will be approached from a computational point of view, asking how might motion correspondence be successfully established.

The range of possible correspondence strategies is determined, to a large degree, by the level at which the matching process is carried out. That is to say, the correspondence strategies to be considered depend on whether the correspondence is performed by matching high-level constructs such as perceived objects [Warren, 1977; Ullman, 1977b], or by matching low-level units, such as primal sketch elements [Marr, 1976], or even individual intensity points as suggested by Anstis [1970]. The evidence in Ullman [1977a] supports the view that motion correspondence is established primarily between low-level units such as points, blobs, edge-fragments, line segments, and certain groups thereof. If this view is correct, the understanding of the correspondence process between such elements should provide an adequate basis for the theory of motion correspondence in general.

We shall start by addressing the correspondence problem in a simplified version, in which two "frames" are presented in succession, resulting in "apparent" motion. We further assume that each frame consists only of isolated points of equal intensitiy. Subsequently, we shall extend the analysis to other types of elements, and to continuous motion.

In looking for a plausible correspondence procedure, we shall use guidelines from "above" and constraints from "below". The guidelines from "above" are properties of physical motion which can be used to solve the correspondence problem. The constraints from "below" are those imposed by computational considerations in general, and by the requirement that the computation will be feasible on a biological system in particular.

2. The Optimal (independent) correspondence strategy

Given the two frames, the problem we face is how to establish a correspondence between their elements. Assuming there are n elements in each frame, there are $n!$ different one-to-one mappings between them. Hence we face an ambiguity problem common to various aspects of visual analysis (e.g., the stereo match problem [Marr and Poggio, 1976], the analysis of occluding contours, [Marr 1977], the interpretation of structure from motion, [Ullman 1977a]). Namely, that the visual input admits more than a single interpretation. In the face of such an ambiguity no method is guaranteed to always yield the correct interpretation. However, if the structure of the task domain renders some interpretations more likely than others, it becomes possible to select the most likely solution, thus maximizing the probability of interpreting the input correctly. We shall therefore look for a correspondence scheme that will maximize the probability of yielding a correct interpretation.

The selection of the most plausible correspondence requires the utilization of information concerning the plausibility of different matches. Such additional information can belong to one of two categories: general or particular. In using particular information, one brings to bear knowledge applicable to a specific situation, e.g. assuming that the black blob on the desk in one's office is a telephone. Examples of general knowledge are the rigidity constraint in the interpretation of structure from motion, [Ullman, 1977a] which is based on properties of rigid objects in general, or the two constraints governing the stereo match [Marr and Poggio, 1976]. If motion correspondence is established at a low level, then information of the general kind should be applied. In the following section properties of moving elements in general will be used to guide the matching problem.

The independence hypothesis

The selection of the most likely correspondence requires a way of comparing the likelihood of different possible matches. To determine the likelihood of a match, one needs to know what dependencies are assumed to hold between the motions of individual elements. For example: if X and Y are neighboring points and X moves to the right, is Y more likely to move to the right than to the left? Since our prime objective is the

investigation of human motion perception, we want our underlying assumptions to be consistent with the correspondence process as carried out by the human visual system. When the human correspondence process is examined using simple displays containing a small number of elements [Ullman, 1977a], no such biases are apparent.

Consider, for example, the configuration in Figure 1 where points *X1* and *X2* are presented in apparent motion with *Y1*, *Y2* and *Y3*. (In all the figures unfilled circles will denote the first presentation and filled circles the second. A filled circle inside an unfilled one means that this element participated in both the first and the second frame.) If only *X1*, *Y1* and *Y2* are presented, *X1* moves to the right or to the left with equal probabilities. It will usually split and move in both directions at once. When *X2* and *Y3* are presented as well, *X2 is seen to move to the right* and match *Y3*. Will this motion increase the likelihood of seeing *X1* as moving to the right to match *Y2*? The answer is that (provided that fixation is maintained at the center) no such preference is apparent.

We will generalizing from this observation and additional evidence from [Ullman, 1977b], and accept the hypothesis that the elements are treated as moving independently of each other. Given this independence hypothesis, we shall next turn to develop the optimal correspondence strategy. It will subsequently be shown that the emerging method remains optimal under conditions which violate the independence hypothesis, and that the incorporation of dependencies between directions would be redundant.

The maximum likelihood correspondence

Suppose that n elements are moving in space independently of each other, at various speeds, and in different directions. Two "snapshots" of the moving elements are taken, and a match is to be established between the "input elements" in the first image and the "output elements" is the second. Let $\rho(v)$ denote the probability density of the velocity distribution of the elements. That is to say, if a moving element is selected at random, the probability that its velocity v lies between values a and b is:

$$\int_a^b \rho(v)\, dv.$$

Assuming spatial isotropy (i.e., that the elements have equal probabilities of moving in any direction in space), then the most likely match is determined by the function $\rho(v)$ in the following way. Let d denote the distance (in the image plane) covered by a given element during some time interval t. If no depth information is used at this stage, we can only conclude that the average velocity of the element in space was at least $v_0 = d/t$. (This expression holds for parallel projection. The change required for perspective projection is insignificant.) If the element moved parallel to the picture plane, its velocity must have equalled v_0, otherwise it was higher than v_0. The probability of an element covering a distance d in time interval t is therfore given by its probability of traveling at a speed v_0 or higher, which is given by the "tail integral" p(v):

(1) $$p(v) = \int_{v_0}^{\infty} \rho(v)dv$$

Given the independence hypothesis, the probability of having a collection of n elements, with the i^{th} element ($1 \le i \le n$) covering a distance d_i in time interval t is given by the product:

(2) $$\Pi\, p(v_i) \qquad v_i = d_i/t$$

The most likely match will therefore be found by maximizing (2) over all the legal matches (the one-to-one mappings in this case). In what follows it will be convenient to transform the product in (2) into a sum. Since the Logarithmic function is monotonic, and since all the $p(v_i)$ are positive, the most likely match can equivalently be found by:

(3) $$\min \Sigma\, q(v_i)$$

where the minimum is taken over all the legal matches, and $q(v) = -\log p(v)$. Since $0 \le p(v) \le 1$, q(v) is a non-negative function. If q(v) is thought of as a "cost" function, then the optimal mapping minimizes the cost over all the legal matches.

Mappings which are not one-to-one

If the number of input and output elements is not equal the mapping between them cannot be one-to-one. The simplest example of this situation is depicted in figure 2 where *A* is presented in apparent motion with both *B1* and *B2*. The one-to-one condition has to be violated in this case, and this can happen in one of two ways. Either *A* is mapped with a single element, leaving the other without a "partner", or *A* can split and match both *B1* and *B2* (Or, if *B1* and *B2* precede *A*, the two elements might both match *A*, a situation we shall call "fusion".) Perceptually, the latter possibility is preferred (unless one of the distances is much larger then the other, [Ullman, 1977a]).

We shall therefore assume that legal matches are required to be *covers*. A cover is defined as a match in which every input element is paired with at least one output element, and every output element is paired with at least one input element. How should the optimal match be determined in these non one-to-one cases? The independence hypothesis as formulated above does not apply directly to situations in which elements split and fuse. For the sake of simplicity, we shall extend the independence hypothesis to include covers as well. Some further modifications of the optimal mapping will be introduced after a method for computing optimal matches has been presented. For the present. the optimal match will be determined, as before, by minimizing $\Sigma\, q(v_i)$ over all the legal matches. The only change is that the set of legal matches is extended to include all covers. In graph theoretical terms the optimal match defined in this way is called the "minimum weighted cover of a bipartite graph". (In a bipartite graph the set of vertices $V = V_1 \cup V_2$, $V_1 \cap V_2 = 0$, and each arc connects a vertex in V_1 to a vertex in V_2.) However, for brevity's sake, we shall refer to the match determined by $\min\Sigma\, q(v_i)$ as the "mΣ mapping".

We next turn to examine the constraints from "below", namely computational problems associated with determining the optimal match.

3. Computational feasibility

The mΣ solution is obviously computable. For instance, by enumeration: the sum $\Sigma\, q(v_i)$ can be computed for all the legal matches, and then a minimum can be selected. However, due to its inefficiency, such an algorithm would be unreasonable. Is there a feasible method of computing the mΣ mapping? The feasibility of a computation depends, to a large extent, on properties of the processor which carries it out, and therefore the question cannot be settled without making some assumption about the way the computation is carried out. Without committing ourselves to a particular model, we shall make three general assumption about the way the correspondence process is carried out by the human visual system. We shall then investigate whether the computation of the mΣ method is feasible given these assumptions. The three assumptions are parallelism, simplicity, and locality.

Parallelism: Since the correspondence process operates on low-level elements and since there might be a large number of those in a given image, the pairing of corresponding elements is probably carried out to a large extent in parallel.

Locality: If the number of "processors" is large, it becomes unfeasible to connect each one of them to all the others. It will threfore be assumed that there are only local connection between the processors, e.g., each processor is connected only to its k nearest neighboring processors. The number k will be called the "radius of the computation".

Simplicity: If there is a large number of processors, it seems reasonable to assume that each of the individual processors is a rather simple computing device. We shall not attempt here to define simplicity precisely, but will only assume that the processors have no memory.

We shall combine the above assumptions in the notion of a *network of simple processors*. All the processors are identical, and each one is connected to k of its neighbors. In the correspondence computation, each processor is "assigned" to an element in the image, and its task is to find a match for this element.

We have listed some theoretical reasons for assuming that motion correspondence is carried out by a simple, local process. A further reason for considering such a computation is that the analysis in [Ullman, 1977a] supports the view that the correspondence process used by the human visual system is indeed simple and local.

General issues such as computability, efficiency and locality in simple networks of this kind are yet little understood. However, rather then addressing them directly, we shall restrict our discussion to their relation to the correspondence process. Since the mΣ method had been advanced as an optimal matching strategy, and simple networks as a plausible computational model, the main problem addressed in this section is: *can the mΣ method be computed by a simple network?* (It should be noted that such simple networks are not, in general, equivalent to a universal computing machine.)

The prospects of performing the mΣ computation with a simple network might seem dubious due to the discrete, combinatorial character of the problem. However, we shall see that the computation can be carried out if the problem is changed somewhat. Instead of the set of all covers we consider a subset of local covers. For each element there are N neighbors which are the *initial candidates* for a legal match. A legal match is one where each element is paired with (at least) one of its initial candidates. Of these legal matches, the one that minimizes $\Sigma q(v_i)$ is sought. We shall verify in the following section that in this formulation the optimal match is computable by a simple local process. We shall also determine the radius of the computation, that is, to how many neighbors must each processor connect to make the mΣ computation possible. *As it turns out, it* is sufficient that each processor be connected only to its initial candidates (i.e., $r = N$).

4. Computing mΣ by a simple, local, network.

In this section we shall present a method by which a simple network can compute the most likely match. The development involves two stages:

1. Reformulating the computation of mΣ as a Linear Programming (LP) problem. A theorem from Integer Programming (IP) ensures the equivalence of the original *problem* and the LP formulation.
2. Employing a method devised by Arrow *et al* [Arrow, Hurwicz & Uzawa; 1958] to solve the resulting LP problem by a simple, local, process.

Reformulating mΣ as a LP problem.

Linear Programming (LP) is the study of optimizing linear functions subject to linear constraints. In vector notation, an LP problem is:

(4) Minimize $\underline{c} \cdot \underline{x}$
Subject to: $A\underline{x} \geq \underline{b}$
$\underline{x} \geq 0$

Where $A = (a_{ij})$ is an n*m matrix, $\underline{x}$ and $\underline{c}$ are n-dimensional vectors, and $\underline{b}$ an m-dimensional vector. In a more explicit form, find a vector $\underline{x} = (x_1, x_2, \ldots, x_n)$ that will minimize $\sum_i c_i x_i$ subject to m constraints on the x_i's. The j^{th} constraint is: $\sum_k a_{jk} x_k \geq b_j$, and $x_i \geq 0$ for $i = 1, \ldots n$.

To recast the mΣ problem in terms of LP we shall introduce the variables x_{ij}, $1 \leq i \leq n$ (if there are n input elements) $1 \leq j \leq k$ (if there are k output elements). If an input element i is paired with an output element j, then $x_{ij} = 1$, otherwise $x_{ij} = 0$. In a cover, $\sum_j x_{ij} \geq 1$ for every i, and $\sum_i x_{ij} \geq 1$ for all j. We shall therefore formulate the following LP problem:

(5) Minimize $\Sigma\, x_{ij}\, q_{ij}\, x_{ij}$

Subject to: $\Sigma_j\, x_{ij} \geq 1$ for $1 \leq i \leq n$

$\Sigma_i\, x_{ij} \geq 1$ for $1 \leq j \leq k$

$x_{ij} \geq 0$ for $1 \leq i \leq n$, $1 \leq j \leq k$

Comments: 1.) The total number of variables x_{ij} is nN, since there are n input elements, each having N neighboring output elements. 2.) q_{ij} is the cost of the link between input element i and output element j.

Is this LP problem equivalent to the original $m\Sigma$ problem? It would be, if we add the restriction that each x_{ij} can assume binary values only (i.e. $x_{ij} = 1$ or $x_{ij} = 0$). The additional restriction cannot be expressed in the LP formalism, but fortunately it is redundant. A theorem from Integer Programming states that there exists an optimal solutions to the above LP problem in which all all the x_{ij} are *integers*. [Garfinkel & Nemhauser; 1972. Note that the constraints matrix is unimodular.] It is straightforward to verify that the integer condition implies that the only possible values for the x_{ij} in the optimal solution are 0 or 1. Consequently, if the optimal solution is unique, any algorithm that solves the LP problem is also guaranteed to solve the original $m\Sigma$ problem. If the optimal soution is not unique, then there are at least two different optimal integer solutions, and also non-integer solutions. For the present, we shall assume that the optimal solution is unique. The non-unique case is examined in section 6.

We shall next describe a method of solving the LP problem which can be carried out by a simple local network.

Computing $m\Sigma$ in a simple network

A method of optimizing functions in which the computation is distributed between simple, locally connected processors, was introduced by Arrow *et al* [Arrow, Hurwicz & Uzawa; 1958]. This method is based on a theorem by Kuhn and Tucker [1951] which states the equivalence between optimal solutions to the constrained problem, and saddle points of the associated Lagrangian.

Consider the problem of maximizing a function f(x) subject to m constraints $g_i(x) \geq 0$, $i = 1,\ldots,m$. The *Lagrangian* associated with the problem is defined as:

(6) $L(x,u) = f(x) + \Sigma\, u_i g_i$

Where x is n-dimensional vector and u is m-dimensional vector. A non-negative saddle point of the above Lagrangian is a non-negative point (x',u') satisfying:

(7) $L(x,u') \leq L(x',u') \leq L(x',u)$ for every $x \geq 0$, $u \geq 0$.

Theorem: (Kuhn - Tucker) If (i) f(x) and $g_i(x)$ are concave, and (ii) there exists a vector $x_0 \geq 0$ such that $g_i(x) > 0$ ($1 \leq i \leq m$), then a vector x' is a solution to the maximization problem if and only if there exists a vector u' such that (x',u') is a saddle point of the associated Lagrangian.

(In the above formulation the conditions are slightly different from those in the original Kuhn-Tucker theorem. For proof, see Arrow, Hurwicz and Uzawa, ch. 3.)

The Lagrangian gradient method

The Kuhn - Tucker theorem which is an extension of the classical Lagrange Multipliers theory, transforms the problem of optimizing a constrained function to the determination of a saddle-point of the associated Lagrangian. Arrow *et al* [1958] investigated the possibility of computing saddle points using gradient methods. A gradient method searches for a saddle-point of L(x,u) by moving in the direction of the local gradients ("uphill" in x, "downhill" in u), without violating the non-negativity conditions on the variables. This search is defined in terms of the Arrow-Hurwicz differential equations (p. 118):

(8)

$\dot{x}(t) = Lx_i$ if $x_i > 0$

$\dot{x}(t) = 0$ if $x_i = 0$ and $Lx_i < 0$

$\dot{u}(t) = -Lu_i$ if $u_i > 0$

$\dot{u}(t) = 0$ if $u_i = 0$ and $Lu_i > 0$

Where Lx_i is the partial derivative of the Lagrangian with respect to x_i, and Lu_i with respect to u_i.

An approximation to the Arrow-Hurwicz equation can be defined by the following iterations: [Marr and Poggio, 1976]

(9)

$x_i^{n+1} = \max [0, x_i^n + \rho\, Lx_i]$

$u_i^{n+1} = \max [0, u_i^n - \rho\, Lu_i]$

where ρ is a selected step size.

If L in the formulae is the Lagrangian as defined in the Kuhn-Tucker theorem, the method is called the "naive" Lagrangian method. The main point to notice is that the naive gradient computation of the $m\Sigma$ problem is simple and local. The reason for the locality is that the values of Lx_i and Lu_i are given in terms of the values of x and u in the i^{th} processor and its N immediate neighbors only. More specifically, the Lagrangian is:

(10) $L(x,u) = \Sigma\, q_{ij}x_{ij} - \Sigma_i\, u_i(1 - \Sigma_j\, x_{ij}) - \Sigma_j\, u_j(1 - \Sigma_i\, x_{ij})$

If there are n input and k output elements then $i = 1,\ldots,n$ and $j = 1,\ldots,k$. The derivatives take the simple form:

$Lx_{ij} = q_{ij} + u_i + u_j$

$Lu_i = \Sigma_j\, x_{ij} - 1$

$Lu_j = \Sigma_i\, x_{ij} - 1$

Since the derivatives are local, the process defined by (9) is simple and local.

Convergence:

The Arrow-Hurwicz method is said to converge to a solution if (x(t), u(t)) approach a saddle point of L(x,u) as $t \to \infty$. The naive gradient method as defined above is not guaranteed to converge to a solution. If L(x,u) is linear in both x and u, the solution might go instead into a limit-cycle. However, the naive

Lagrangian method can be modified in a way that will ensure the convergence of the Arrow-Hurwicz equations to a saddle-point (of both the modified and the original Lagrangian), and hence to an optimal solution.

The modified Lagrangian LM is defined as [Arrow *et al*, p. 137]:

(11) $LM(x,u) = f(x) + \Sigma u_i \psi_i[g_i(x)]$

where the functions ψ_i are strictly increasing, strictly concave analytic functions with $\psi_i(0) = 0$ (An example of such a function is $\psi(z) = 1 - e^{-rz}$ for $r > 0$.)

Note that the gradient method applied to the modified Lagrangian still yields a local computation, similar to the naive gradient case.

If the iterative procedure in (9) is applied to this modified Lagrangian, the iteration will usually converge to a solution as well. Furthermore, it is also possible to modify the original Lagrangian in such a way, as to guarantee the global convergence of the iterative procedure to a solution provided that the step-size ρ is sufficiently small, while maintaining the locality property of the procedure. As before, the gradients LMx and LMu computed for the i^{th} component will depend only on the i^{th} processor and its N immediate neighbors.

Conclusions

The optimal match mΣ can be determined by a simple, local computation. One can envision a network of simple processing elements which accepts two "snapshots" of elements in motion, and finds the most likely correspondence between them via local interactions. The above conclusion can be applied to other problems of constrained optimization, for details see [Ullman, 1978].

5. Preference for one-to-one mappings

The mΣ method as presented above does not "penalize" matches for deviating from the one-to-one mapping. Such a *simplification is unsatisfactory on both theoretical and empirical* grounds.

On the theoretical side, splits and fusions of elements in real images are unlikely, though not impossible, e.g. in the case of one element occluding another in one of the snapshots. Let δ denote the probability of such an occlusion. That is, the probability of a simple split (an input element splitting to link with two output elements) or a simple fusion (two input elements converging onto the same output element) is δ. The probability of an element having three links ("double occlusion") is δ^2. In general, the probability of a split with $s + 1$ links is δ^s, and the probability of a fusion with $f + 1$ links is δ^f. The probability of a match containing k splits with $s_1+1,\ ...,s_k+1$ links, and n fusions with $f_1,\ ...,f_n$, is given by:

(12) $\Pi\ p(v_i)\ \delta^{s_1} ... \delta^{f_n}$

By taking the -log of the above expression we get that the "cost" of the match is (where $\sigma = -\log \delta$):

(13) $\Sigma q(v_i) + \sigma(\Sigma s_i + \Sigma f_j)$ $i = 1,\ ...,k$, $j = 1,\ ...,n$.

The optimal match is found by minimizing (13). The larger the σ in this last expression (that is, the smaller the probability of splits and fusions), the higher will be the preference for one-to-one mappings.

There are empirical grounds as well for associating additional penalty with splits and fusions. Figure 3 provides an example. The match in figure 3a (*A1→B1←A2, B2→A3←B3*) minimizes Σ qi (this statement holds for high ISI, see section 6) but the one-to-one match in figure 3b (*A1→B1, A2→B2, A3→B3*) is perceptually preferred.

We shall next see how to modify the mΣ method so that it minimizes the penalized sum in (13).

The modified mΣ method

Rather than minimizing $\Sigma q_{ij}x_{ij}$, let us now minimize $\Sigma q_{ij}x_{ij} + k\Sigma x_{ij}$ $(= \Sigma(k + q_{ij})x_{ij})$.

As before, in the optimal solution the x_{ij} will be binary, hence the "penalty function" $k\Sigma x_{ij}$ is simply k times the total number of links in the match. By making k larger, mappings with smaller number of links will be preferred. Furthermore, the next proposition shows that for the appropriate choice of k we can minimize the required sum in (13).

Proposition: Minimizing $\Sigma q_{ij}x_{ij} + 2\sigma\Sigma x_{ij}$ (subject to the usual constraints $\Sigma x_{ij} \geq 1$) is equivalent to minimizing $\Sigma q_{ij} + \sigma(\Sigma s_i + \Sigma f_j)$ over all covers.

Proof: First note that chains of corresponding elements are precluded. For examine the chain: *A1 → B1 ←A2 →B2*. The link *B1 ← A2* can be removed without violating the constraints hence this chain cannot be a part of the optimal solution. Let m be the number of one-to-one links in a given match. The total number of links in this match is:

(14) $\Sigma x_{ij} = m + \Sigma(s_i + 1) + \Sigma(f_j + 1)$ where i ranges over the splits and j over the fusions.

The number of input elements I is given by:

(15) $I = m + \Sigma|s| + \Sigma(f_j + 1)$ where $|s|$ is the total number of splits. The number of output elements O is given by:

(16) $O = m + \Sigma(s_i + 1) + |f|$ where $|f|$ is the total number of fusions.

We now subtract $\sigma(I + O)$ from the objective function. This quantity does not depend on the match, therefore it does not alter the minimization problem (a match minimizes the penalized sum $\Sigma q_{ij}x_{ij} + 2\sigma\Sigma x_{ij}$ if and only if it minimizes $\Sigma q_{ij}x_{ij} + 2\sigma\Sigma x_{ij} - \sigma(I + O)$). By substituting for Σx_{ij}, I, and O, the penalty $2\sigma\Sigma x_{ij} - \sigma(I + O)$ becomes:

(17) $\sigma(\Sigma s_i + \Sigma f_j)$

Minimizing $\Sigma q_{ij}x_{ij} + 2\sigma\Sigma x_{ij}$ is therefore equivalent to minimizing $\Sigma q_{ij}x_{ij} + \sigma(\Sigma s_i + \Sigma f_j)$. Since the x_{ij} are binary, (and constrained by $\Sigma x_{ij} \geq 1$) this is equivalent to minimizing $\Sigma q_{ij} + \sigma(\Sigma s_i + \Sigma f_j)$ over all covers. ∎

Note that optimizing the penalized sum does not affect the computation. The cost q_{ij} can subsume the constant k, so that

the optimal solution is still found by minimizing $\Sigma\, q_{i,j}x_{i,j}$. The computation thus remains simple and local while exhibiting the required degree of preference for one-to-one mappings.

6. Properties of the mΣ mapping

So far we have characterized the optimal correspondence strategy by a certain mathematical condition, namely minimizing a cost function over all the local covers. In this section we turn to examine some of the properties of the mΣ mapping and to compare them, when possible, to properties of the correspondence established by the human visual system.

Minimizing the total distance

As the iter-stimulus interval (ISI) between successive frames increases, the mΣ mapping is expected to minimize the total distance covered by all the elements in the image. If d_i is the distance traveled by the i^{th} input element, then the mΣ method will minimize the quantity $\Sigma\, d_i$ over the legal matches.

Proof:

We shall make the further assumption that the probability of low velocities is approximately constant. This assumption seems reasonable: while very high velocities are unlikely, there is no reason to assume that a velocity of, say 1 deg/second is considerably more (or less) frequent than a velocity of, say, 0.5 deg/second. Near the origin, the function $\rho(v)$ can therefore be described as $\rho(v) = k$ for some constant k. In the region where this approximation holds, the functions p(v) and q(v) assume the form:

$$(18)\quad p(v) = \int_v^\infty \rho(u)\,du = 1 - kv$$

$$q(v) = -\log(v) \sim kv \quad (\text{since } kv << 1)$$

Minimizing $\Sigma\, q_i$ is hence equivalent in the case of low velocities (or high ISI between the frames) to minimizing $\Sigma\, v_i$ or, equivalently, (since $v_i = d_i * ISI$) to minimizing $\Sigma\, d_i$. ∎

The rule of non-crossing trajectories

It has been noticed [Kolers, 1972; Attneave, 1974; Navon, 1976] that the paths of elements in apparent motion seldom cross. If *A1*, *A2*, are shown in apparent motion with *B1*, *B2*, in a configuration where the paths *A1* → *B2*, *A2* → *B1* cross but *A1* → *B1*, *A2* →*B2* do not (see figure 4), then the latter match is preferred (provided that the ISI is sufficiently large).

The rule of non-crossing trajectories is implied by the minimum distance principle. The triangle inequality implies that $(d_1 + d_2) < (c_1 + c_2)$ (in figure 4). That is, the non-crossing trjectories always minimize the total distance and therfore, for high ISI, also minimize Σq_i.

Flow detection

Suppose that two snapshots (S1 and S2) are taken of a collection of elements moving parallel to each other. We shall refer to such a parallel motion as a *flow* of the elements. The visual system seems capable of detecting flows: when the two snapshots S1 and S2 are presented in succession, the flow motion will usually be perceived (provided that the ISI is not too short). This holds true even when the average distance traveled by the elements between the two snapshots is considerably larger than the mean inter-elements distances, in which case most of the elements are not paired with their nearest neighbors.

This flow detection capacity deserves a closer examination since it appears not to be consistent with the independence hypothesis made in section 2. It seems to indicate that each element "prefers" a match whose direction is consistent with the direction of neighboring elements. The independence hypothesis, on the other hand, excluded interactions based on direction similarity. The flow detection phenomenon might also suggest the existence of some global measurements, which do not belong to any single processor in the simple network discussed in section 4. The prevailing orientation can be discovered by a global measurement and can then affect the match assigned to the individual elements. However, such a suggestion concerning interactions between local and global processes runs contrary to the simple network model. The flow detection phenomenon therefore raises the following problem: In a simple network model, the correspondence between collections of elements is governed completely by the local interactions. According to the independence hypothesis, these local interactions do not include positive interactions between matches of similar directions. Yet, when a common direction does exist it seems to affect the correspondence process, as indicated by the flow detection phenemenon.

To resolve this difficulty, we shall turn to examine the flow detection in the light of the mΣ mapping. The conclusion we shall reach is that flow detection is not at odds with either the independence assumption or the simple network model. In fact, it supports them since, as we shall see, the mΣ method actually implies flow detection.

Recall that S2 is obtained from S1 by translating all the elements along a common direction. The correct match between S1 and S2 is the one in which each element in S1 is paired with its translated image in S2. We now wish to establish:

Claim (the flow-detection lemma):

The correct match minimizes the total distance $\Sigma\, d_i$ (over all the one-to-one mappings).

Proof:

Let (x_i, y_i) denote the position (in the image plane) of the i^{th} input element, and (x'_i, y'_i) its position in the second snapshot. If the X-axis is chosen to coincide with the direction of the flow, then $y'_i = y_i$, and $x'_i \geq x_i$. A match between the snapshots is a function m which assigns an output element to every input element. Thus $j = m(i)$ means that the j^{th} output element is paired with the i^{th} input element.

The total distance Dc of the correct match is given by:

$$(19)\quad Dc = \Sigma\,(x'_i - x_i), = \Sigma\, x'_i - \Sigma\, x_i.$$

For another match m, the total distance Dm is given by:

$$(20)\quad Dm = \Sigma\,[\,(x'_j - x_i)^2 + (y'_j - y_i)^2\,]^{1/2} \qquad i = 1,\ldots,n \quad j = m(i).$$

$$(21)\quad Dm \geq \Sigma\,|x'_j - x_i| \geq \Sigma\,(x'_j - x_i) = \Sigma\,(x'_i - x_i) = Dc \qquad i = 1,\ldots,n \quad j = m(i).$$

Since $Dm \geq Dc$, Dc is minimal, and the correct match is optimal. ∎

It can also be seen that Dm will be strictly greater than Dc unless m is also a flow, namely $y'_j = y_i$, $x'_j \geq x_i$. Outside some special situations the optimal match will therefore be unique.

The flow-detection lemma can also be proven for the case of radial motion. Suppose that each element moves (in the image plane) along the line which connects it to a certain fixed point o. (Such radial flow can arise from an approaching objects, as well as from the perspective projection of pure translation in space). Then, the correct radial correspondence minimizes the total distance.

Proof: Let o be the origin, and describe the position of each element by its polar coordinates (r, θ). If d_{ij} is the distance between input element i and output element j, $d_{ij} \geq |r_i - r_j|$. For a given match m,

$$(22)\quad Dm = \Sigma d_{ij} \geq \Sigma |r_i - r_j| \geq \Sigma (r_i - r_j) = Dr$$

$$(i = 1,\ldots,n;\quad j = m(i))$$

Where Dr is the total distance of the correct (radial) correspondence. ∎

The independence hypothesis revisited

The optimal correspondence strategy $m\Sigma$ has been developed for independently moving elements. The independence assumption might be questioned on the ground that proximate elements in the image are likely to move in similar directions. It can be argued, therefore, that if a "locally parallel" match (i.e., a match in which the motion of proximate elements is nearly parallel) exists it should be preferred. While there is probably some truth to this, the flow detection analysis suggests that the explicit incorporation of such a preference might be redundant, since parallel motions also minimize Σd_i. The $m\Sigma$ mapping is thus a plausible method whether or not the motion of the elements is indeed independent. It should also be noted that the $m\Sigma$ methods requires only the measurement of distances between elements but not of directions, a property that might have an advantage in terms of economical implementation. Since the number of elements in a scene can be large. a computation of the optimal correspondence based on a minimal number of parameters, and with a minimal number of interactions, might offer an important advantage.

Symmetry

One property of the human correspondence process is "a preference for symmetrical movement, more important things being equal" [Attneave, 1974, p. 118]. Such a symmetry property is to be expected in any simple, local network as defined in section 4. Furthermore, if there is a symmetry in the input, then there must be a symmetric optimal match. Symmetry is defined as a permutation π that "does not alter the problem". That is to say, if q_{ij} is the cost of the link between input element i and output element j, and q'_{ij} is the cost of the link between $\pi(i)$ and $\pi(j)$, then for all i and j, $q_{ij} = q'_{ij}$ If such a symmetry exists, then:

1). There exists a symmetrical optimal match, i.e., a match in which $x_{ij} = x'_{ij}$. This holds because if (x_{ij}) (a sequence of 1's and 0's) is an optimal solution, then so is (x'_{ij}). The solution (y_{ij}) defined by $y_{ij} = (x_{ij} + x'_{ij})/2$ is also optimal, and symmetric.

2). The iterative procedure (9) will converge to a symmetric optimal solution. Since the input is symmetric, the first stage in the iteration is symmetric. Since all the processors are identical, the next stage, and by induction all stages, will be symmetric too. The symmetric configurations can be divided into two categories: integer and non-integer. Figure 5 exemplifies a non-integer symmetric configuration. Figure 5a shows one optimal mapping and figure 5b another. The mapping in Figure 5c is a combination of the two, and is both optimal and symmetric. As has been noted in section 4, when (and only when) the optimal solution is not unique, there exist also non-integer optimal solutions, of which figure 5c is an example. The mapping in figure 5c is expected to be unstable, since it relies on the exact equality of the distances *A2-B1*, and *A2-B2*. Any deviation from the strict equality between these distances will cause either figure 5a or figure 5b to be optimal. It is not surprising, therefore, that the perception associated with this configuration is unstable and alternates between the two [Kolers, 1972; Attneave, 1974; Ullman, 1977a].

Figure 6 shows an optimal, symmetric, integer mapping. Unlike the non-integer mappings, these are perceptually stable. This stability is not completely predictable from the $m\Sigma$ method since it depends on properties of the algorithm by which the method is carried out. It can be verified that if a row of n elements is shown in alternation with a row comprising $n+1$ elements, then whenever n is even the symmetric solution is non-integer, and whenever n is odd there exists a symmetric, optimal, integer solution. It is therefore reasonable to expect that in the first case the perceived match will be unstable and asymmetric, and in the second symmetric and stable. This prediction is consistent with the observations of Kolers [1972] and of Attneave [1974].

Symmetry in the order of presentation

When two frames f1 and f2 are shown in apparent motion, the perceived correspondence does not depend on the order of presentation. That is, the pairing of elements remains the same whether f1 precedes or follows f2 [Ullman 1977a] This symmetry is shared by the $m\Sigma$ correspondence process. The optimal solution to the matching problem remains invariant when the input and output elements switch roles.

The minimal cover property

The $m\Sigma$ mapping is a minimal cover in the sense that it does not contain superfluous links. The removal of any link from the match will result in one input or output element "uncovered" (i.e., without a counterpart). This property implies the phenomena of split and fusion competition discussed in Ullman [1977a]. Figure 7 explains the split competition. In figure 7a, element *A1* is presented followed by a pair of flanking elements *B1* and *B2*. *A1*

is perceived as splitting and matching both *B1* and *B2*. In figure 7b, a second element, *A2*, is added to the first frame. The resulting correspondence is *A1* → *B1*, *A2* → *B2*, while the link *A1* → *B2* disappears. It is as if *A2*, by taking over *B2*, competes with *A1* and prevents it from matching *B2*. In the mΣ mapping the three links *A1* → *B1*, *A1* → *B2*, *A2* → *B2*, cannot co-exist since this maping will not be a minimal cover (*A1* → *B2* is removable). Similarly, Attneave [1974] described configurations in apparent motion where the number of links is kept to the minimum required to supply each element with a partner. Observing this minimal cover property as well as such properties as symmetry and non-crossing paths, Attneave [1974] commented:

"It would appear that the system is exhibiting foresight, and one is strongly tempted to invoke some *deus ex machina*, some superordinate, ratiomorphic control system that makes everything come out neatly" [Attneave, 1974, p. 116]

The discussion in the preceeding sections suggests that no such global planning is required. A simple, local process can possess all the discussed properties, and is in fact expected to exhibit them if it computes the mΣ mapping.

Monotonicity in the rate of sampling

Roughly speaking, if the minΣ d_i mapping yields the correct correspondence when the second view is separated by time interval t from the first, then for every t' < t the match will also be correct.

To prove the claim, we shall assume that the elements are moving along straight lines (an assumption which will hold for short time intervals). Two snapshots of the moving elements are separated by time interval t are given. Suppose that the correct match (i.e., the match in which each input element is paired with the same element after the time interval t) minimizes Σ d_i. Then, for every t' < t, the correct correspondence will also minimize Σ d_i.

Proof:

Let v_i be the velocity of the i^{th} element, σ will denote the total distance Σ d_i of the correct match at time t, and σ' the total distance at time t'. Let μ' be the total distance Σ d_i of some one-to-one match m' at time interval t'. We wish to establish that $\sigma' \leq \mu'$. From the match m' at time t' one can obtain a match m at time t: If a point x is paired by m with some point y(t') (i.e., point y at time t'), then m is obtained by pairing x with y(t). The correct match is y(0) → y(t') → y(t). In m' x(0) → y(t'), and in m x(0) → y(t). We shall denote by μ the total distance of the new match m. From the assumption that the correct match minimizes Σ d_i at time t, $\sigma \leq \mu$. To prove our claim, it is therefore sufficient to show that $\sigma - \sigma' \geq \mu - \mu'$. The contribution of element y to σ is $v_y t$ and its contribution to σ' is $v_y t'$. Let the difference between the two contributions be $r = v_y(t - t')$. In match m x → y(t) and in m' x → y(t'). The difference between the two contributions of y is (d - d'), and (d - d') ≤ r (the *triangle inequality*). Similar inequalities hold for all the elements, hence $\mu - \mu' \leq \sigma - \sigma'$. Combined with the known inequality $\sigma \leq \mu$, the implication is that $\sigma' \leq \mu'$, that is, σ' is minimal. ∎

The monotonicity property has possible application for the correspondence computation. For example, in eliminating wrong matches by checking for consistency with intermediate matches. Suppose that an element x in S1 is matched with y in a subsequent frame S2, and z in a third frame S3. If the correspondence is correct, the matches must be consistent, i.e., y → z. By accepting only consistent matches, the correspondence process can reduce the number of wrong matches. Such a consistency check can be performed for any correspondence scheme, regardless of monotonicity. However the monotonicity implies that "false alarms" in which x → z is correct but the match is rejected, are highly unlikely. Observations of the human coorrespondence process suggest that the human visual system does not use such consistency verifications. However, it seems that, in accordance with the monotonicity property, the performance of the human correspondence apparatus improves monotonically with the rate of sampling.

The shape of q(v) and some of its implications

The preference for nearest neighbors:

It has been frequently noted that the human correspondence process tends to match each element with its nearest neighbor, whenever such a choice is possible without violating other conditions. For example, in figure 8a, element *Z* can be paired with either *Y1* or *Y2*. Both matches will be legal, since in both each of the input elements (*X*, *V*, *Z*) is paired with at least one output element, and each of the output elements (*Y1*, *Y2*) is paired with at least one input elements. In such a situation the matching of *Z* with its nearest neighbor will always be preferred. That is, if d1 < d2 in figure 8a, the match *Z* → *Y1* will be preferred over *Z* → *Y2*. In figure 8b, on the other hand, *Z* will match *Y2*, since a match with its nearest neighbor *Y1* will produce an illegal match.

The preference for the nearest neighbor might seem to suggest that the correspondence method incorporates the assumption that lower velocities are more probable than higher ones. We have noted, on the other hand, that in the low velocity range a more plausible expectation is that all velocities are about equally probable. If this latter view is correct, what might account for the strong preference for nearest neighbors at all velocities?

The answer is that, in the framework of the mΣ method, the nearest neighbor should be preferred regardless of the probability distribution of velocities in the environment. Recall that q(v), the function to be minimized by the correspondence process, was defined as -log p(v), where p(v) is the "tail" integral $\int_v^\infty \rho(u)du$.

Since $\rho(u) \geq 0$, p(v) is monotonically decreasing in v regardless of the shape of ρ. A correspondence process which minimizes q(v) should therefore prefer nearest neighbors even if, for example,

high velocities were actually more probable than low ones.

The convex region of q(v)

The following relation holds between the shapes of q(v) and the underlying distribution $\rho(v)$. Where $\rho(v)$ is either constant of increasing, q(v) is convex. Where $\rho(v)$ decreases, q(v) can assume any shape: convex, concave or linear. (for $\rho(v) = ke^{-kv}$, q(v) is linear). It has been assumed (section 6) that for low velocities $\rho(v)$ is roughly constant. In this region q(v) is therefore convex.

Implication to Ternus' configuration:

Ternus' configuration in apparent motion [Ternus, 1926; Pantle & Picciano, 1976; Ullman, 1977a] is composed of two dots (*A* and *B*) presented in brief succession with a second pair (*B*, *C*). Dots *A*, *B*, and *C* lie on the same horizontal row (figure 9). Depending on various conditions, the perceived correspondence can be in one of two modes. In the "coherent" mode the pair (*A*, *B*) moves as a unit to the right (i.e., the perceived correspondence is $A \rightarrow B$, $B \rightarrow C$). In the "neighbor" mode $B \rightarrow B$, while *A* often "jumps over" to match *C*.

In the convex region of q(v) the mΣ mapping implies the coherent mode of correspondence. In the coherent mode the distances of the match are both equal to d. In the neighbor mode, one of the distances is 0, the other is 2d. The convexity of q(v) implies that:

(22) $q(0) + q(2d) > 2q(d)$

Hence, the coherent mode minimizes Σq_i.

The concave region of q(v)

At high velocities $\rho(v)$ decreases and q(v) is no longer necessarily convex. Existing data suggest that at the high velocity region q(v) is concave. The function q(v) thus assume a sigmoid shape, as diagrammed in figure 10.

Implication to Ternus' configuration:

The sigmoid shape of q(v) implies that when v is sufficiently large, the mΣ method will prefer the neighbor over the coherent mode. Figure 5c depicts the transition point between the two modes, where $q(0) + q(2d) = 2q(d)$. Note that if the element *B* is displaced by a distance h to the left between presentations, the total distance of the coherent mode decreases (by 2h) while the total distance of the neighbor mode remains unchanged. As predicted by the mΣ mapping, in this version of the Ternus configuration the preference for the coherent mode increases with h.

The "non-crossing paths" rule re-examined

In section 6 we have suggested that the rule of non-crossing trajectories is merely a reflection of minimizing Σq_i under low velocity conditions. If this view is correct, and in the light of the shape of q(v), one can expect the rule to break under specified conditions. In Figure 4 the match $A1 \rightarrow B1$, $A2 \rightarrow B2$, minimizes the total distance Σd_i and is therefore expected to prevail under low velocity conditions. However, at high velocities (e.g. short ISI conditions), the sigmoid shape of q(v) implies that the other match, in which $A2 \rightarrow B1$, should prevail. This is indeed the case, in contrast to the rule of non-crossing paths.

7. The experimental determination of q(v)

The optimal match between two collections of points is by the functions q(v). (q(v) can include the constant k of the modified mΣ method.) If the visual system incorporates a correspondence method similar to the mΣ mapping, can we discover experimentally the function q(v) used by the visual system? Before outlining a way of investigating q(v), we shall examine the following question: can q(v) be determined *uniquely* by examining the matches established by the visual system? Suppose that a function q'(v) exists, which always predicts the same matches as q(v) (i.e., $\Sigma q'_i$ is minimal whenever Σq_i is). Such a function q'(v) would be indistinguishable from q(v). However, it is possible to show that, q(v) can in principle be determined up to a scaling factor. If only one-to-one mappings are examined, q'(v) is indistinguishable from q(v) if and only if $q'(v) = cq(v) + b$ for some constants b and c. That is, by examining one-to-one matches, q(v) can be determined up to a linear function. The following procedure is an example of how q(v) can be so determined. We shall make use of bistable displays, similar to the Ternus' configuration. If a bistable configuration has two equally probable matches *m* and *m'*, then $\Sigma q_i = \Sigma q'_i$, where Σq_i is the total cost of *m* and $\Sigma q'_i$ of *m'*. In the Ternus configuration, when the transition between the modes occurs, then:

(23) $\hat{q}(0) + \hat{q}(2v_1) = \hat{q}(v_1) + \hat{q}(v_1)$

Let us arbitrarily set $\hat{q}(0)$ to 0, and $\hat{q}(v_1)$ to 1. Consequently, $\hat{q}(2v_1) = 2$. The notation $\hat{q}(v)$ rather than q(v) has been used to draw a distinction between the function $\hat{q}(v)$ (which is determined by the bistable configurations with $\hat{q}(0) = 0$ and $\hat{q}(v_1) = 1$) and the true function q(v) that we are after.

We can now use v_1 and $v_2 = 2v_1$ to determine new values of $\hat{q}(v)$. In Figure 7 we can change v_3 selectively while maintaining v_1 and v_2 fixed, until a bistable configuration is reached (i.e., $A1 \rightarrow B1$, $A2 \rightarrow B1$, and $A1 \rightarrow 2$, $A2 \rightarrow B1$, are equally probable). When this condition is reached, then $\hat{q}(v_1) + \hat{q}(v_2) = \hat{q}(v_3) + \hat{q}(v_3)$. Hence, $\hat{q}(v_3)$ is also determined (Figure 7b). Theoretically, this method can be extended to determine q(v) on a dense set of values (i.e. between any to known values it is possible to get another value). The function $\hat{q}(v)$ can therefore be measured. We now come back to our original function q(v), which is a linear function of $\hat{q}(v)$, that is $q = a\hat{q} + b$. To determine the additive constant we can use bistable configurations in which the total number of paired elements is different in the two possible matches. Figure 7 is an example of such a configuration. By gradually increasing the distance y while keeping all the other distances constant, a bistable situation will be reached in which:

(24) $q(v_1) + q(v_1) + q(v_2) + q(v_2) = q(v_1) + q(v_1) + q(v_3)$

Substituting $q = a\hat{q} + b$, we get: $a\hat{q}(v_2) + b + a\hat{q}(v_2) = a\hat{q}(v_3) + b$

$\hat{q}(v_2)$ and $\hat{q}(v_3)$ are already known, so b/a is determined as well.

Since q can be determined only up to a scaling factor, we conclude that $q = c(\hat{q} + b/a)$ where c is an arbitrary constant.

8. Extensions

The discussion thus far has concentrated on the correpondence between two frames, containing points of equal intensity. In this section the notion of seeking the most likely match between elements via a simple local process, will be extended to include various types of elements and continuous motion.

Extending the set of elements

As mentioned in the introduction, the set of basic elements matched by the correspondence process includes such units as edge fragments, line segments, and blobs. The main novelty introduced by extending the set of basic elements is that the optimal mapping is no longer determined by time intervals and distances alone. The likelihood of a match between two elements is influenced in the general case by other parameters, such as orientation, length, and contrast. These parameters influence the likelihood of the match between two given elements, and therefore they enter the correspondence process via the "cost" function q. However the optimal mapping still minimizes $\Sigma(q_{ij} + k)x_{ij}$, and can be determined by the local network discussed in section 4.

Some empirical evidence supports the view that the match selected by the human correspondence process can indeed be predicted on the basis a "cost function" with the following properties: (1) It is weighed by orientation, length, and intensity as well as by distance. (2) The relative effect of the various parameters is consistent with likelihood considerations.

Examples of (1):

The likelihood of crossing trajectories (section 6) increases if the elements across the diagonal, but not along the sides, are identical. Similarly, one can favor selectively the neighbor or the coherent mode in Ternus' configuration by mainipulating the similarity (in terms of orientation, length, and intensity) of the participating elements.

Example of (2):

Changes in length and orientation of small line segments in the image are induced primarily by rotation in space (perspective effects are of secondary importance for small segments and short time intervals). When a segment rotates in the image-plane its length remains unaltered. If it rotates in depth, its orientation is unchanged but its length decreases. If α is the angle of rotation, the ratio of final-to-original length in this last case is cos (α). If matches are selected on the basis of likelihood, and given space isotropy (i.e., rotations in every direction are equally probable), the effect on the preferred match of α degrees orientation difference and cos (α) length ratio should be comparable. The data in Ullman [1977a] are in close agreement with this prediction.

Continuous motion

The goal of this section is to extend the analysis from the discrete presentation of two frames to continuous motion. We shall see that the optimal solution can be established in the continuous case as well by a simple, local process. The network that carries out this computation is a simple extension of the one described in section 4, and reduces to it in the case of discrete presentation.

In the continuous case time varies continuously, but we assume that the location of the elements does not. Namely, elements can be detected at discrete locations in the image. Unlike the discrete case, the appearance and disappearance of the elements at different locations is no longer synchronized. We shall consider the case of n elements moving about between times $t = t_0$ and $t = T$. As an introduction to the general case we shall make the assumption that at $t = t_0$ and $t = T$ all n elements are present in the image.

The legal matches in this case are the following. Each of the n elements at $t = t_0$ has one link connecting it to a later element (i.e., an element that appears at a later time). Each of the n elements a $t = T$ has one link connecting it to an earlier element. Each intermediate element has two links, one to an earlier, the other to a later element. By the independence hypothesis, the optimal match is the one that minimizes Σq_i over the legal matches (where i ranges over all the links in the match).

In the two frames situation the correspondence was equivalent to a cover problem on a bipartite graph, with the bipartite structure playing an important role. We shall now formulate the continuous correspondence as well in terms of covering a bipartite graph. We shall view each element as a pair, composed of a "source" and a "sink". The sources are responsible for establishing connections with later elements, the sinks with earlier elements. Each source has as its initial candidates all the later sinks within a certain spatial neighborhood. The graph of possible pairings now becomes bipartite, the set of all sources being one component, and the set of all sinks the other. As before the optimal match can be found by:

(25)

Minimize $\Sigma q_{ij}x_{ij}$

Subject to $\Sigma x_{ij} \geq 1$ for every i, where i ranges over all the sources, and to

$\Sigma x_{ij} \geq 1$ for every j, where j ranges over all the sinks.

As before the problem so formulated is equivalent to the optimal correspondence problem provided that $x_{ij} = 1$ if the i^{th} source is matched with the j^{th} sink, and $x_{ij} = 0$ otherwise. Since on a bipartite graph x_{ij} are guaranteed to be binary, the formulations are equivalent. It is also possible to bias the optimal match towards a minimal number of connections by replacing q_{ij} by $(q_{ij} + k)$ as was done previously. The problem so formulated is formally identical to that of section 4. Hence, the local process (in equations 8 and 9) will converge to the optimal match.

In the continuous motion correspondence the cost function q depends not only on the elements and their spatial separation, but also on their separation in time. As might be expected, for the visual system q_{ij} (the cost of the link between elements i and j) increases with the time interval that separates them. Other parameters (including velocity) being equal, the match which minimizes separation in time will be preferred. The likelihood of a match between a pair of elements, which is inversely related to the cost q, decrease with the time interval separating the elements. If this time interval exceeds some upper limit τ, the two elements are no longer considered candidates for a match. Rather than having a common time interval within which correspondence is established (the interval t_0 - T in the previous example), each element has as potential matches only the elements within a time interval τ In such a network there is no "first" or "final" snapshots; the optimal correspondence is computed continuously as the input elements are streaming in.

REFERENCES

Anstis, S. M. 1972. Phi movement as a subtraction process. *Vision Research, 10, 1411-1430.*

Arrow, K. J., Hurwicz, L., and Uzawa, H. 1958. *Studies in Linear and Non-linear Programming.* Stanford: Stanford University Press.

Attneave, F. 1974. Apparent motion and the what-where connection. *Psychologia, 17,* 108-120.

Garfinkel, R. S. and Nemhauser, G. L. 1972. Integer Programming. New York: Wiley & Sons.

Kolers, P. A. 1972. *Aspects of Motion Perception.* New York: Pergamon Press.

Kuhn, H. W. and Tucker, A. W. 1951. Nonlinear Programming. In J. Neyman (ed.) *Proc. of the Second Berkeley Symp. on Math. Stat. and Prob.* Berkeley and Los Angeles: University of California Press, 481-492.

Marr, D. 1976. Early processing of visual information. *Phil. Trans. of the Roy. Soc of Lond. B, 275 (942),* 483-524.

Marr, D. and Poggio, T. 1976. Cooperative computation of stereo disparity. *Science 194,* 283-287.

Marr, D. 1977. Analysis of occluding contour. *Proc. R. Soc. Lond. B, 197,* 441-475.

Navon, D. 1976. Irrelevance of Figural Identity for Resolving Ambiguities in Apparent Motion. *J. of Exp. Psychol. Vol 2,(1)* 130-138

Ullman, S. 1977a. The interpretation of visual motion. *Ph.D. Thesis, M.I.T., Department of Electrical Eng. and Comp. Science.*

Ullman, S. 1977b. Transformability and object identitiy, *Perception & Psychophysics, Vol. 22.(4),* 414-415.

Ullman, S. 1978. Simple networks in visual information processing. *In preparation.*

Warren, H. W. 1977. Visual information for object identity in apparent motion. *Perception & Psychophysics, 21,* 264-268.

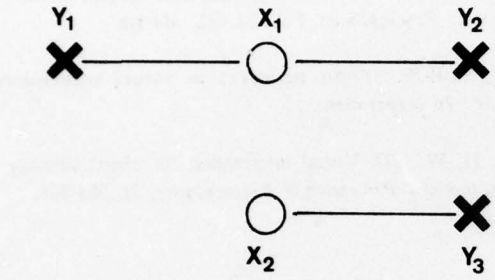


FIGURE 1

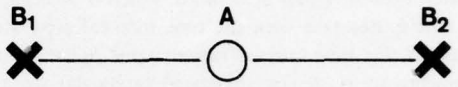


FIGURE 2

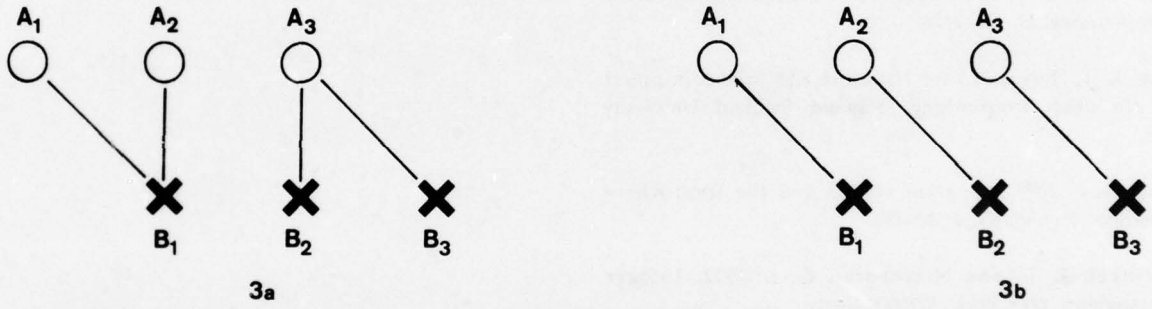


FIGURE 3

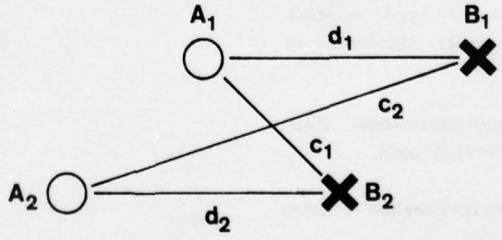


FIGURE 4

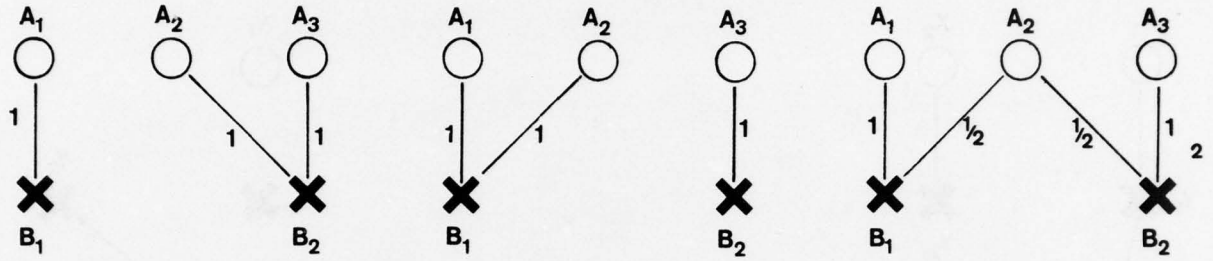


FIGURE 5

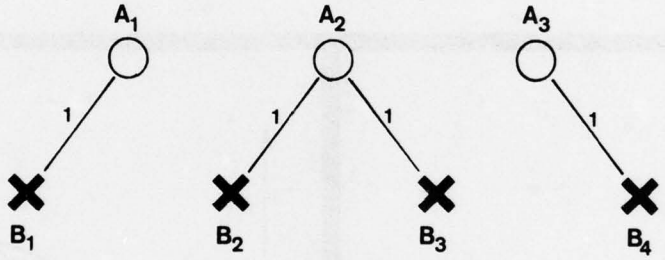
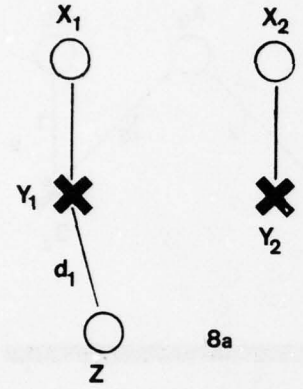


FIGURE 6

7a 7b

FIGURE 7



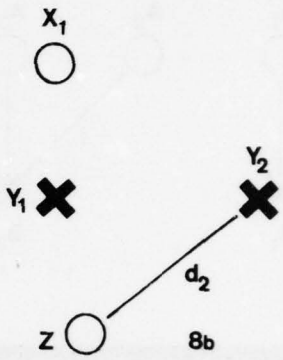


FIGURE 8

FIGURE 9

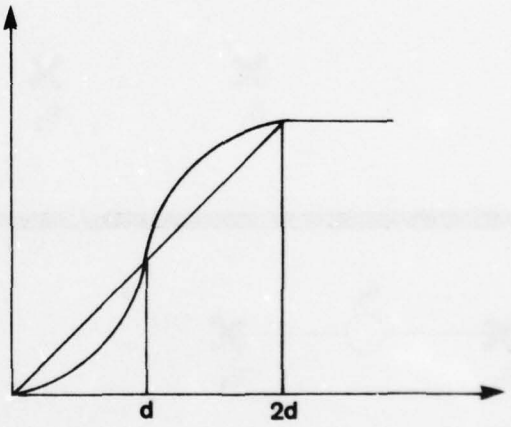


FIGURE 10

A Semantic-Syntactic Approach to Image Understanding And Creation

G. Y. Tang and T. S. Huang
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907
U.S.A.

Abstract

We apply semantic grammar to image understanding and creation. Understanding refers to the problem of recognizing a given pattern. Creation refers to the searching for a pattern amid a chaos of primitives.

Two examples are given.

I. INTRODUCTION

We propose the injection of semantic features into a context free grammar [1,2] for the purpose of image analysis.

A feature vector is assigned to each terminal and to each nonterminal. A feature transfer function (which can be algorithmic) is attached to each production rule. The feature transfer function transfers features at the right hand side of the production rule to the left hand side nonterminal. Following Knuth [1], we call his augmented grammar a semantic grammar. It is very similar but not identical to the attribute grammar [3] [5].

We have applied the semantic grammar to two image analysis tasks: understanding and creation. By understanding we mean the recognition of a given pattern. By creation we mean the searching for patterns amid a chaos of primitives.

The task of pattern understanding is accomplished as follows. After syntactic parsing, the feature vector associated with the root of the derivation tree is sent to a discrimination function to determine the semantic well-formedness of the sentence. More generally, at any intermediate parsing stage the feature vector of a nonterminal can be checked and the pattern rejected if the feature vector does not meet a prespecified criterion - in particular, we can impose selection restrictions [6]. The acceptance of an input signal is thus based on not only its syntactic structure but also its semantic contents.

To do creation, the semantic grammar is used as a guide to control the searching processs. For example, we may want to search for long straight line segments amid a chaos of edge points detected by some local operator. To do that, we first develop a semantic grammar for long straight line segments. Then this grammar is used to aid the search. At any stage of the search, which edge point to look at next is suggested by the appropriate production rules of the grammar.

In this paper, the application of semantic grammar to a one-dimensional pattern understanding problem will be described in detail. Then the results of a two-dimensional problem involving creation will be presented. However, the details of this second example are omitted because of the lack of space.

II. A ONE-DIMENSIONAL EXAMPLE

We use semantic grammar to recognize highways and edges in aerial photos. The grey level distribution along a straight line segment crossing the highway or edge is obtained by a film scanner.

The a priori knowledge about the signal is that it looks like one of the four paradigms α, β, γ, σ in Fig. 1. α, β are paradigms for the ideal edges. γ, σ are the paradigms for highways.

The grammar describing the ideal paradigms is:

```
1  : 0 → D̂ A B   ; F1
2  : 0 → D̂ B A   ; F1
3  : 0 → D̂ A     ; F7
4  : 0 → D̂ B     ; F7
5  : A → a X     ; F2
6  : X → D̂       ; F3
7  : X → D̂ A     ; F6
8  : B → b Y     ; F2
9  : Y → D̂       ; F3
10 : Y → D̂ B     ; F6
11 : A → a       ; F8
12 : B → b       ; F8,
```

0 is the start symbol, a, b, $\hat{D}$ are terminals, A, B, X, Y are nonterminals.

The transformation, which brings the ideal paradigms to the realistic level, is to replace each occurrence of the symbol $\hat{D}$ by a sentence generated by the grammar:

```
T 1 : D → f D     ; F4
T 2 : D → c D1    ; F4
T 3 : D → d D2    ; F4
T 4 : D → f       ; F5
T 5 : D → c       ; F5
```

T 6 : D → d ; F5

T 7 : D_1 → d D_2 ; F4

T 8 : D_1 → d ; F5

T 9 : D_1 → f D ; F4

T10 : D_2 → c D_1 ; F4

T11 : D_2 → c ; F5

T12 : D_2 → f D ; F4

T13 : D_1 → f ; F5

T14 : D_2 → f ; F5

D is the start symbol, and c, d, f are terminals.

The transfer functions associated with the production rules are defined as:

F1 : $\underline{A} \rightarrow \underline{B}\ \underline{C}\ \underline{D}$

$\tilde{W}(\underline{A}) = \tilde{C}(\underline{D}) - \tilde{C}(\underline{C})$

$\tilde{C}(\underline{A}) = \tilde{C}(\underline{C})$

$\tilde{R}2(\underline{A}) = 2$ if $\tilde{R}2(\underline{C}) \neq 0$ and $\tilde{R}2(\underline{D}) \neq 0$
0 otherwise

$\tilde{A}(\underline{a}) = |\tilde{A}(\underline{D}) - \tilde{A}(\underline{C})|$

$\tilde{R}1(\underline{A}) = \text{Max}(\text{Max}(\tilde{R}1(\underline{C}), \tilde{R}1(\underline{D})), \tilde{A}(\underline{B})/\tilde{W}$

F2 : $\underline{A} \rightarrow \underline{B}\ \underline{C}$

$\tilde{A}(\underline{A}) = \tilde{A}(\underline{B}) + \tilde{A}(\underline{C})$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{B}) + \tilde{W}(\underline{C})$

$\tilde{C}(\underline{A}) = \tilde{C}(\underline{B})$ if $\tilde{C}(\underline{C}) = 0$

$(\tilde{C}(\underline{B}) + \tilde{C}(\underline{C}))/2$ if $\tilde{C}(\underline{C}) \neq 0$

$\tilde{R}1(\underline{A}) = \tilde{R}1(\underline{B})$

$\tilde{R}2(\underline{A}) = \tilde{R}2(\underline{B})$

F3 : $\underline{A} \rightarrow \underline{B}$

$\tilde{A}(\underline{A}) = \tilde{A}(\underline{B})$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{B})$

$\tilde{R}1(\underline{A}) = \tilde{A}(\underline{B})/\tilde{W}(\underline{B})$

$\tilde{R}2(\underline{A}) = \tilde{W}(\underline{B})$

$\tilde{C}(\underline{A}) = 0$

F4 : $\underline{A} \rightarrow \underline{B}\ \underline{C}$

$\tilde{A}(\underline{A}) = \text{Max}(\tilde{A}(\underline{B}), \tilde{A}(\underline{C}))$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{B}) + \tilde{W}(\underline{C})$

$\tilde{C}(\underline{A}) = 0$

F5 : $\underline{A} \rightarrow \underline{B}$

$\tilde{C}(\underline{A}) = 0$

$\tilde{A}(\underline{A}) = \tilde{A}(\underline{B})$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{B})$

F6 : $\underline{A} \rightarrow \underline{B}\ \underline{C}$

$\tilde{A}(\underline{A}) = \tilde{A}(\underline{C})$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{C})$

$\tilde{R}1(\underline{A}) = \tilde{R}1(\underline{C})$

$\tilde{R}2(\underline{A}) = 0$ if $\tilde{W}(\underline{B}) \geq t$

$\tilde{R}2(\underline{C})$, otherwise

$\tilde{C}(\underline{A}) = \tilde{C}(\underline{C})$

F7 : $\underline{A} \rightarrow \underline{B}\ \underline{C}$

$\tilde{R}1(\underline{A}) = \text{Max}(\tilde{R}1(\underline{C}), \tilde{A}(\underline{B})/\tilde{W}(\underline{B}))$

$\tilde{C}(\underline{A}) = \tilde{C}(\underline{C})$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{C})$

$\tilde{R}2(\underline{A}) = 1$ if $\tilde{R}2(\underline{C}) \neq 0$
0 otherwise

F8 : $\underline{A} \rightarrow \underline{B}$

$\tilde{C}(\underline{A}) = \tilde{C}(\underline{B})$

$\tilde{W}(\underline{A}) = \tilde{W}(\underline{B})$

$\tilde{A}(\underline{A}) = \tilde{A}(\underline{B})$

$\tilde{R}2(\underline{A}) = 1.$

The underscored symbols like $\underline{A}$ are formal parameters. Symbols with ~ atop designate features associated with the formal parameters in parentheses, e.g $\tilde{C}(\underline{A})$ is the $\tilde{C}$ feature attached to $\underline{A}$.

There are five terminals, a, b, c, d, f. To each terminal, there are three features attached. Literally a, b, c, d, and f are five tendencies in the input signal. a and c represent the tendency of going-up. b and d are for going-down. f is for flatness. The extent of going-up differentiates a from c. a stands for long going-up. c stands for short going-up. Similarly b is long going-down and d is short going-down.

The three feature attached to the terminals are $\tilde{A}$, $\tilde{W}$, and $\tilde{C}$. $\tilde{C}$ refers to the center of the tendency. $\tilde{W}$ refers to width of the tendency. $\tilde{A}$ is a measre of the opposition (long/short). $\tilde{A}(a) = 1$ or $\tilde{A}(b) = 1$ means absolutely long. More specifically, let $\&(\cdot)$ denote the height of the tendency. Then for a, b, we have

$$\tilde{A}(S) = (\ell(S)/M_1 - t)/(1-t).$$

For c, d, we have

$$\tilde{A}(S) = [\ell(S)/M_1 - t)/t] + 1.$$

M_1 is the maximum height. $t\,M_1$ is the threshold for discriminating between "long" and "short".

For nonterminals, there are two more features. These two features are defined by the transfer functions.

The final semantic well-formedness test is:

$$\tilde{W}(0) \geq t_1\ ,\ \tilde{A}(0) \leq t_3$$

$$\tilde{R}1(0) \geq t_2$$

$$\tilde{R}2(0) > 0.$$

$\tilde{C}(0)$ is the location of the edge or the leading edge of the highway. $\tilde{W}(0)$ is the width. $\tilde{R}(0) = 1$ indicates edges. $\tilde{R}(0) = 2$ indicates highways. t_1, t_2, and t_3 are preset thresholds.

Experimental Results

An aerial photograph is shown in Fig. 2. The gray level along the white straight line segments are used as the input signals. Each of the 5 input signals (one for each segment) is parsed with the grammar described above to determine whether it contains a highway (a single edge will be rejected).

Correct answers were obtained for all five cases. Two examples are shown in Figs. 3 and 4. Fig. 3 shows the gray level variation along segment #3 of Fig. 2. A highway is recognized. Fig. 4 shows the gray level on variation along segment #5 of Fig. 2. No highway is found here.

III. A TWO-DIMENSIONAL EXAMPLE

We use semantic grammar to look for airplanes in the photo shown in Fig. 5. The task is accomplished in three steps.

First, a local edge detector was used to obtain edge points as shown in Fig. 6. Then, semantic grammar for long straight line segments was used to search for such items in the edge-point picture. Finally, a semantic grammar for airplanes was used to search for airplanes. The airplane is found as shown in Fig. 7. Note that because the semantic grammar was developed for complete airplanes, the partial airplane in Fig. 5 was not detected. The details of the grammars used in this example can be found in a forthcoming technical report [4].

REFERENCES

1. D. E. Knuth, Semantics of Context Free Languages, Math. Syst. Theo. Vol. 2, pp. 127-146, 1968.
2. K. S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, 1974.
3. P. M. Lewis, D. J. Rosenkrantz, and R. Z. Sterns, Compiler Design Theory, Prentice Hall, 1976.
4. G. Y. Tang and T. S. Huang, A Semantic-Syntactic Approach to Image Understanding and Creation, Technical Report, School of Electrical Engineering, Purdue University, 1978.
5. G. V. Bochmann, Semantic Evaluation from Left to Right, Comm. of ACM, Vol. 19, No. 2, Feb. 1976.
6. G. Leech, Semantics, Penguin books, 1976, pp. 141-143.

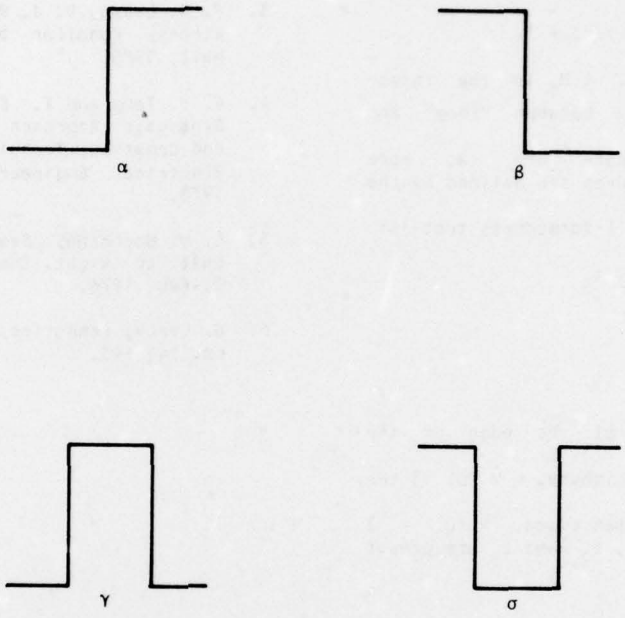


Fig. 1 Four paradgms. α, β, represent edges. γ, σ, represent highways.

Fig. 2 An aerial photograph.

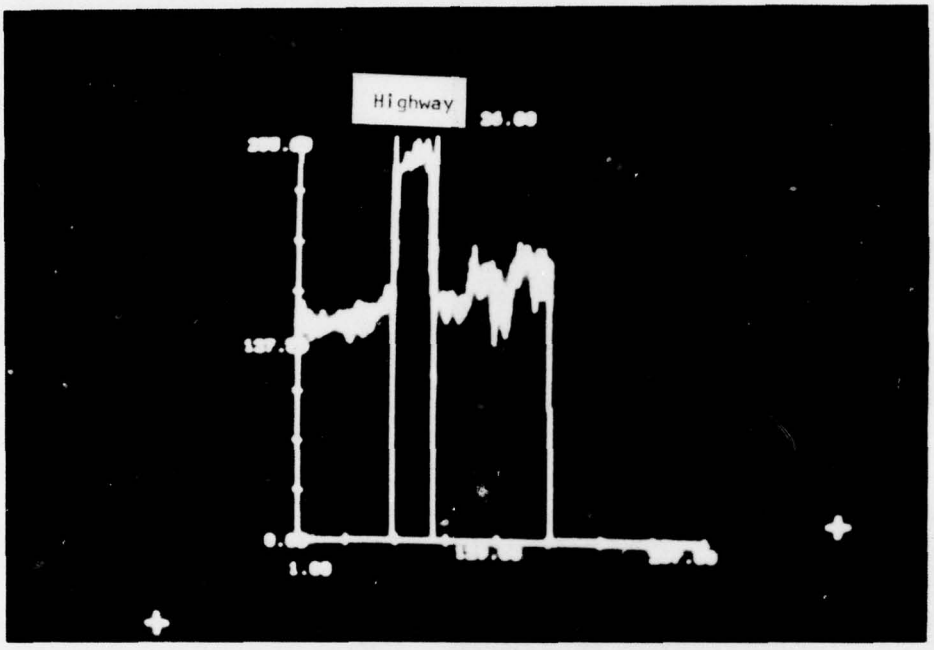


Fig. 3 Highway is bounded by the two vertical lines. Its width is 26 pixels.

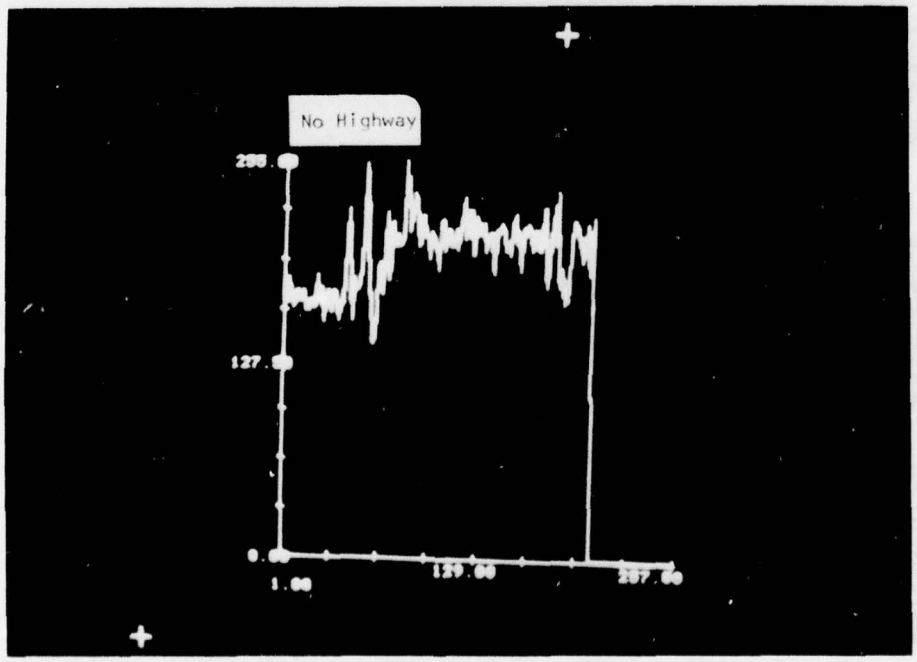


Fig. 4 No highway is found.

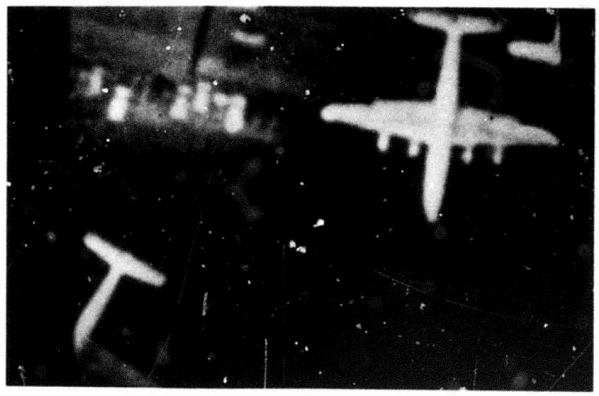


Fig. 5 An aerial photo containing airplanes.

Fig. 6 Edge points of Fig. 5 detected by a local operator.

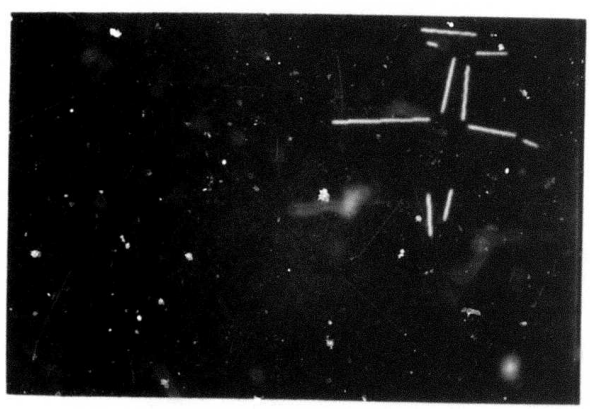


Fig. 7 Airplane recognized.

SYMBOLIC MATCHING AND ANALYSIS WITH SUBSTANTIAL CHANGES IN ORIENTATION

Keith Price

Image Processing Institute
University of Southern California
Los Angeles, California 90007

Abstract

Most previous image matching work has assumed that the two images being matched are already in close alignment or that transformations are given which will closely align the images. This paper shows how symbolic matching techniques can be applied to pairs of images to accurately locate the corresponding objects in the two views when the required transformations are not known a priori. We present two scenes where there are major orientation differences between the two views and show the results of the matching procedure on these scenes.

Introduction

Several methods have been developed which can be used to find correspondences in pairs of images of a changing scene [1-6]. But, for various reasons, these systems did not operate on images which had major changes in orientation - unless an approximate value for the difference was known a priori. Image based methods [1-3] were not used to attempt a solution to this problem because of the complexity of searching for corresponding points. Other work [5,6] assumes that the views are close - in time and position - so that major changes in the point of view are not possible. But in a more general image matching and analysis system [4], this type of problem must be considered.

We have undertaken a series of experiments to examine whether the matching techniques described in [4] can be easily applied to pairs of images which have substantial global differences. In this paper, we present the results of applying this basic technique on pairs of images with orientation differences of 45°, 90° and 180°. These orientation changes are in addition to other, less drastic, changes which may occur between the two views. The point is to show that symbolic techniques can be applied when there are substantial global changes in addition to the normal local changes which occur between two views of one scene.

We will first discuss the images which will be used for this experiment and describe the results which are desired. Then we will present an outline of the symbolic matching procedure [4] and some initial results using this method. The results suggest some modifications which are then described, followed by more extensive results using the modified procedure.

Tasks for Matching

The pairs of images which we will use here have been used earlier [7], but in the previous analyses there were no substantial global changes. We start with a pair of images of a scene taken from slightly different positions, and generate the orientation changes by rotating the digital representations of the second image of the pair. The original first image and the rotated second images are then processed to generate symbolic descriptions (a segmentation into distinct parts plus a description of the segments) which are used by the matching procedures. The details of the segmentation and description are given elsewhere and are not important for this paper [4,7,8].

The basic task to be executed for the images presented here is to find the corresponding regions in the two images when there is a large, but unknown, orientation difference. A by-product of this matching should be some indication of the actual orientation difference. The orientation-dependent features are not used even if they are independent of the rotations used in the experiment (e.g. ratio of area and area of minimum bounding rectangle).

The first pair of images, a house, is shown in Figure 1. These are color images, shown here in black and white, so there are several spectral features available for use in the segmentation, description, and matching operations. Figure 2 gives the segmentations of the original first view and the three rotated second views. There are some

*This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203 ARPA Order No. 3119.

differences between the regions segmented in the first view and each of the second views, and a few differences among the three second views. The images are not square so the display program puts a white band on the right or bottom, depending on which dimension is smaller. Additionally unsegmented areas are displayed as white areas.

The second pair of images is shown in Figure 3. These are side looking radar views and there is only one spectral input, so that a good segmentation is more difficult and the description is less detailed than the preceding house scene. The important changes in this scene are in objects which are too small to be segmented by our general segmentation techniques but might be easily located by special methods. Figure 4 gives the four segmentations which will be used. There are a few large regions segmented in the first view which are not segmented in the second views. These differently textured regions vary in size and appearance between the two views and would not be used in the matching, therefore they were not segmented in the second views. The regions which are segmented are the untextured areas, all uniformly dark, some of which appear the same in both views. The locations of these few regions can be used to determine the global changes and could aid another system in locating the detailed changes.

Matching Procedure Outline

A more detailed description of the matching procedure has been presented elsewhere [4] and will only be outlined here. This matching procedure uses a feature based, symbolic, description of the images. The basic unit in this description is an individual segment generated by an automatic segmentation procedure [8]. These segments are usually regions in the images, but linear features can be described, too. Features which characterize properties such as color, texture, size, shape, position, and adjacencies are used.

The matching procedure is also given an indication of which features are available for matching the current pair of images, and what strength to give to the mismatch using each of these features. For example, some features are not always computed, red and green in a black and white picture, and some are given as more likely to change than others and are thus given less weight in the matching operation. The match procedure computes a rating for the match between two differences of feature values. The weights used in the sum are composed of a normalization factor to make the contribution from each feature approximately equal and a strength factor to account for the different strengths assigned each feature. There is only a small set of possible strength values, currently 3 different values.

Known global changes between the two images can be used to adjust some feature values, such as size, position, and orientation. But, these changes are not given a priori and must be computed from a few initial pairs of matching regions. Thus, the very clearly defined regions should be matched first so that they may be used for calculating some global changes. In this context, clearly defined regions are regions with extreme values for some feature, e.g. largest, brightest, longest, etc.

We will now present results of applying this procedure to the rotated images and discuss what changes are necessary to achieve accurate results.

Initial Results

Figures 5 and 6 show the results obtained with the above matching procedure for the house scene (90° and 180°). In these, and all other figures, the corresponding regions are displayed at the same intensity in the pair of output pictures. Similar results are obtained for 45° and for matching the second image to the first, but the point here is to show some of the problems. The orientation feature adjustment was computed, using the sky or roof, and was also used to get these results, but there are still many errors. Most of the unmatched regions would match to an incorrect corresponding region if a match is attempted.

The major problem is that the location of the region is needed to correctly locate matches for many of the smaller regions. This is especially the case when there are size and shape changes due to segmentation differences, such as in the bush, window, and door regions. If exact camera transformations are known then the locations in one image can be mapped exactly into locations in the other, but this transformation is not known. We are using many features other than position so an appropriate mapping is sufficient to allow the use of the absolute position features. Given 3 pairs of corresponding regions we can compute a transformation which will map coordinates in one image to coordinates in the other by solving 2 sets of 3 equations and 3 unknowns. This transformation is not optimal for all regions in the image and only accounts for rigid, global changes - e.g. rotations and translations. But this transformation does make the position features usable when there are large global orientation changes.

Final Results

Figures 7-9 show the results for the house matching using the computed location transformations - a different transformation is computed for each image pair.

Results are given for matching regions in image 1 with image 2. The results for regions in image 2 with image 1 are similar. In this set of images the sky, lawn, and one of the wall sections are the initial 3 regions used for the transformation computation. The transformations do not rotate the coordinates precisely 45°, 90°, or 180° because of differences in segmentations (the sky and lawn are adjacent to the edge and this causes some changes in size and shape) and a small orientation difference which existed before the large rotations were added. But, the adjustment is accurate enough to use the location feature in the match operation. The results at 45° are less accurate than for the other two, but most of the extra mistakes are accounted for by the greater differences in segmentations (see Figure 2). When two regions in one image correspond to one region in the second image, such as the 2 "bushes" on the right side of the house in the second image appearing as one region in the first image, only one correspondence appears in the output.

Figures 10 and 11 present the results for the radar images. There are very few (i.e. 6) corresponding regions in these two images and two of the pairs have very large size changes. The results for 180° are identical to those for 90° and are not presented. Two of the corresponding pairs may be difficult to see since they are nearly white in the results picture - one is a correct match (the reversed "C" shape in the lower left), and the other is incorrect (a blob near the top right). The reverse "C" region, the river (lower right) and the blob above the river are the three regions used to compute the transformation in this set of images. This scene shows that when these symbolic techniques are applied to scenes with a reduced feature set - no colors and no neighboring regions, accurate results are possible.

Summary

The complete symbolic registration system presented here has the following basic steps:

1. Segment both images of the scene.
2. Generate a feature based description of the segmented images.
3. Find corresponding regions for the most obvious regions.
4. Set orientation and size correction factors, if necessary.
5. Find several corresponding region pairs.
6. Compute an approximate coordinate transformation, if necessary.
7. Using transformed positions, find all corresponding region pairs.

The matching results depend somewhat on the quality of the segmentation, but the results of these experiments show that this symbolic technique can be used to find corresponding areas in pairs of images even when there are major global changes. We would expect similar, or better, results for pairs of images with global scale, position, and color changes. We expect less reliable results for scenes with major global changes in all four (orientation, scale, color, and position) because so few features are invariant to all these changes (e.g. relative size, shape measures, and neighbors). But if a controlling system could provide proper guidance, corresponding regions might be located which would account for each of the global changes, separately. For example, scale changes could be based on matching the largest regions, orientation changes might be based on regions with distinctive shape, and so on. But, primary regions with unusual, or extreme, feature values could be used when there are many global changes. In conclusion, symbolic matching methods can work with major global differences, these differences can be detected, and they can be used to great advantage in later analysis.

Acknowledgements

The author is at the Image Processing Institute, University of Southern California, Los Angeles, California 90007. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203, ARPA Order No. 3119.

References

[1] G.R. Allen, L.O. Bonrud, J.J. Cosgrove, and R.M. Stone, "The Design and Use of Special Purpose Processors for the Machine Processing of Remotely Sensed Data," IEEE Symposium on Machine Processing of Remotely Sensed Data, Purdue University, October 1973.

[2] L.H. Quam, "Computer Comparison of Pictures," Ph.D. Thesis, AIM-144, Stanford University, Stanford, California, May 1971.

[3] H.P. Moravec, "Towards Automatic Visual Obstacle Avoidance," in Proceedings IJCAI-77, Cambridge, Massachusetts, 1977, p. 584.

[4] K. Price and R. Reddy, "Matching Segments of Images," submitted for publication IEEE-TC.

[5] W.K. Chow and J.K. Aggarwal, "Computer Analysis of Planar Curvilinear Moving Images," IEEE-TC 26, 1977, pp. 179-185.

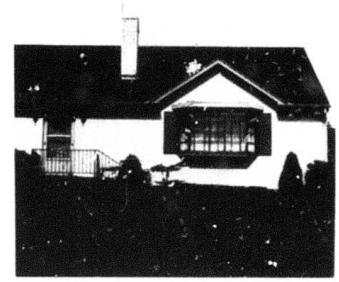
[6] H.H. Nagel, "Formation of an Object Concept by Analysis of Systematic Time Variations in the Optically Perceptible Environment," Computer Graphics and

Image Processing, to appear.

[7] K. Price, "Change Analysis and Detection in Multi-Spectral Images, Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, December 1976.

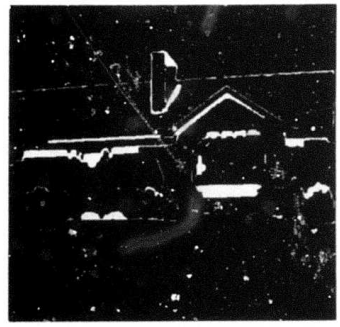
[8] R. Ohlander, K. Price, and R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing, to appear.

(a)

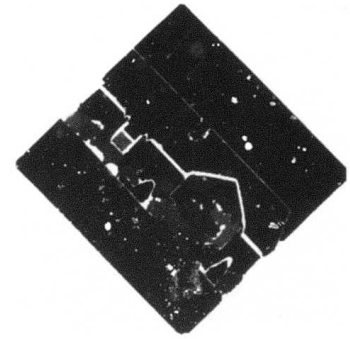


(b)

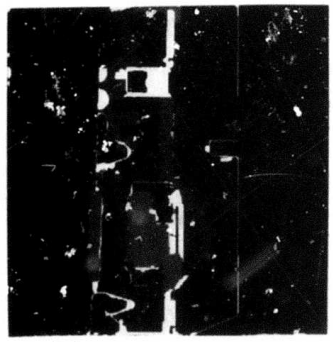
Figure 1. House Images.



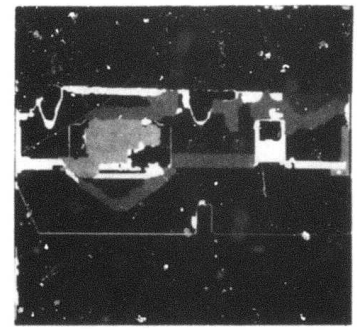
(a) First view



(b) Second view rotated 45°



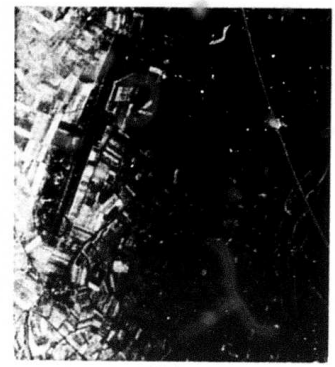
(c) Second view rotated 90°



(d) Second view rotated 180°

Figure 2. Segmentations of House Images.

(a)



(b)

Figure 3. Radar Images.

(a) First view

(b) Second view rotated 45°

(c) Second view rotated 90°

(d) Second view rotated 180°

Figure 4. Segmentations of Radar Images.

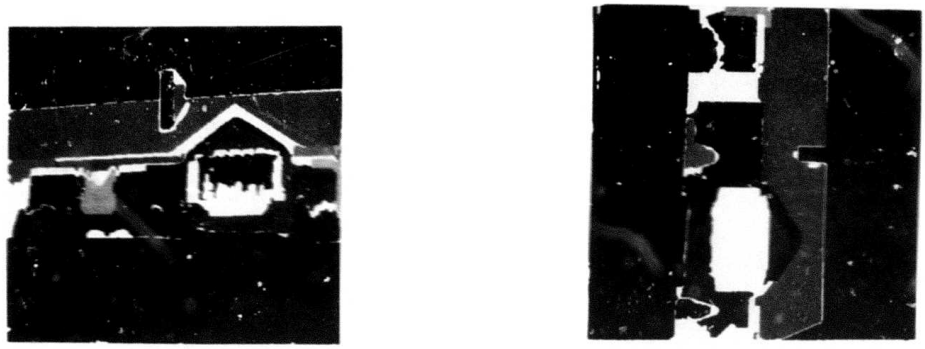


Figure 5. Initial Matching Regions for House First View (left) and House Second View Rotated 90° (right).

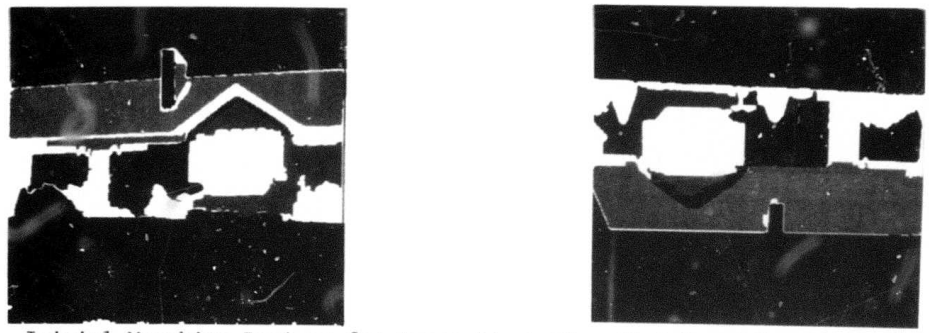


Figure 6. Initial Matching Regions for House First View (left) and House Second View Rotated 180° (right).

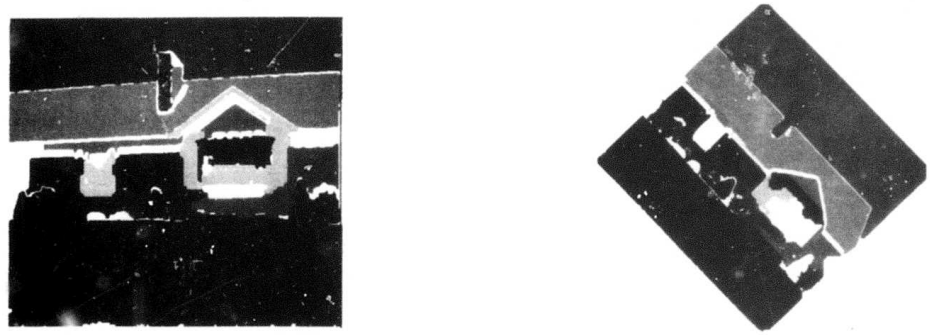


Figure 7. House 1 with House Rotated 45°.

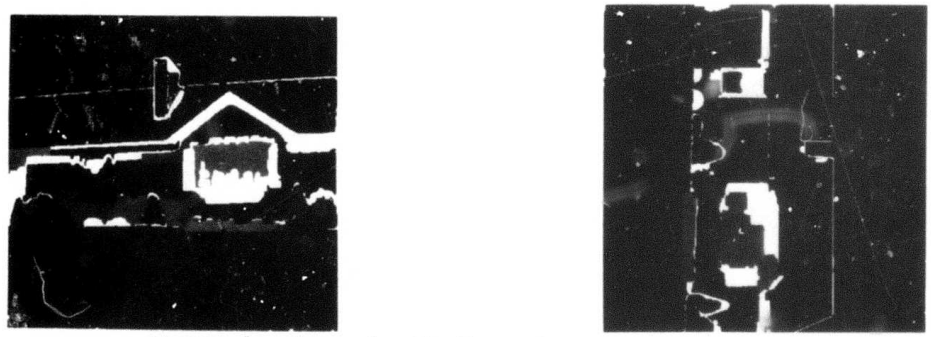


Figure 8. House 1 with House 2 Rotated 90°.

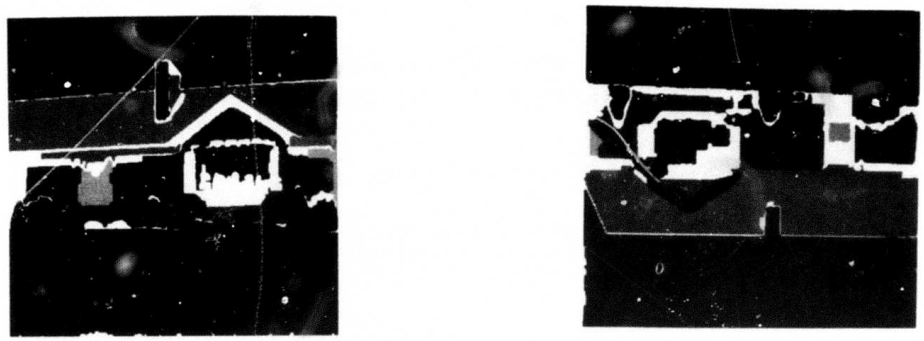


Figure 9. House 1 with House 2 Rotated 180°.

Figure 10. Radar 1 with Radar 2 Rotated 45°.

Figure 11. Radar 1 with Radar 2 Rotated 90°.

SOME RECENT RESULTS USING RELAXATION-LIKE PROCESSES

Azriel Rosenfeld

Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

This paper describes a number of recent experiments involving the use of synchronous iterative processes in low-level computer vision.

INTRODUCTION

Synchronous iterative processes ("relaxation methods") have many potential applications in low-level computer vision. A review of many of these applications can be found in [1], and some further work is summarized in [2]. This paper briefly describes several recent developments.

MULTISPECTRAL PIXEL CLASSIFICATION

Classification of image pixels based on their spectral signatures is commonly used in the analysis of remote sensor imagery, and has also been used to segment other types of color images [3, 4]. The results of this classification are often noisy, since the pixels are classified independently of one another. To reduce the noise, a postprocessing technique can be used; e.g., if most of the neighbors of pixel P have been classified as belonging to class C, then P itself is reclassified as class C.

This postprocessing approach is based on very little information about the pixels; it makes use only of the (most probable) classes to which they were assigned, but not of how close they came to being assigned to other classes. A better informed approach might be to classify each pixel P probabilistically, i.e., to estimate the probability p_i that P belongs to each class C_i, and then to adjust these probabilities based on the class probabilities of the points adjacent to P.

Preliminary experiments have been conducted to compare this probabilistic approach with the simple postprocessing approach described earlier. These experiments made use of red, green, and blue color separations of the house image shown in Figure 1. The image was hand segmented into five regions as shown in Figure 2. The Mahalanobis distance from each pixel to each of these clusters was computed. The initial classification was based on smallest Mahalanobis distance. Figure 3 shows this initial classification. The error rate was 5.6%.

In the postprocessing approach, if a pixel P had six or more neighbors that belonged to class C, P was reclassified as C, and this process was iterated. Figure 4 shows the results of the first and sixth iterations. The error rates are 5.2% and 5.03%, respectively.

In the probabilistic approach, class probabilities were assigned to each pixel P; these were defined by

$$p_i = \frac{1}{4}\left[1 - \frac{d_i}{\Sigma d_j}\right]$$

where d_i is the Mahalanobis distance to the ith cluster mean. These probabilities were then adjusted using the "relaxation" formula of [1-2], with the compatibility coefficients defined by mutual information as in [2]. The errors after eight iterations of the probability adjustment process are shown in Figure 5; the error rate is 1.9%, a major reduction.

For comparison purposes, an iterative preprocessing technique was also used on the same data. This technique is a multispectral analog of one of the noise cleaning schemes described in [1]. Specifically, each pixel's (red, green, blue) color vector was averaged with six of the color vectors of its neighbors -- namely, those six that were closest to it in color space. After this averaging step (which could be iterated), the pixel was classified using closest Mahalanobis distance, as above. The results of this classification, after one and two iterations of the averaging process, are shown in Figure 6. The error rates are 5.35% and 5.04%.

The results of this experiment suggest that the relaxation approach may be more useful than simple pre- or post-

processing in improving the results of multispectral pixel classification. Similar results have also been obtained by E. Riseman. Further experiments using LANDSAT data are in progress.

DETERMINATION OF COEFFICIENTS

In [2] it was shown that reasonable coefficients for a curve enhancement relaxation process can be defined by computing the mutual information between pairs of initial curve probability estimates (at various slopes) at pairs of neighboring points. These initial estimates could be obtained from the given image, or from any image having a reasonable distribution of curve slopes. More recent experiments indicate that usable coefficients can even be obtained by applying curve detectors to a pure noise image. This is because when a detector responds, the probability of a response at a neighboring point, corresponding to a smooth extension of the curve, is far above chance, since the detectors at neighboring points overlap greatly.

A set of coefficients obtained in this way is shown in Figure 7, and results of using them for curve enhancement are shown in Figure 8. These results confirm the idea that the coefficients that should be used to enhance the output of the detector depend on the definition of the detector itself, and not on the statistics of any particular type of input data.

OTHER APPLICATIONS

A number of other experiments using relaxation-like processes are in progress; they are described briefly in the following paragraphs.

a) In the recognition of mechanical parts, pieces of object boundary extracted by a segmentation process can be classified probabilistically as belonging to various portions of a given mechanical part. The class probabilities can then be adjusted, depending on whether or not other portions of the given part appear to be present in the correct (approximate) relative positions.

b) In matching two sketches of a given scene (or in matching a sketch against a segmented image of the scene), feature points on the sketches can be probabilistically paired off based on their similarity. These probabilities can then be adjusted, depending on whether or not other corresponding pairs appear to be present in the correct (approximate) relative positions.

c) In matching two relational structures, nodes can be probabilistically paired off based on their similarity. These probabilities can then be adjusted, depending on whether or not other corresponding pairs appear to satisfy the proper relations with respect to the given pair.

Results on these applications will be reported in a subsequent paper.

REFERENCES

1. A. Rosenfeld, Iterative methods in image analysis, Proc. IEEE Conf. on Pattern Recognition and Image Processing, June 1977, 14-18.

2. A. Rosenfeld, Relaxation methods: recent developments, Proc. Image Understanding Workshop, October 1977, 28-30.

3. R. Ohlander, Analysis of natural scenes, Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, PA, December 1976.

4. B. J. Schachter, L. S. Davis, and A. Rosenfeld, Scene segmentation by cluster detection in color space, SIGART Newsletter, No. 58, June 1976, 16-17.

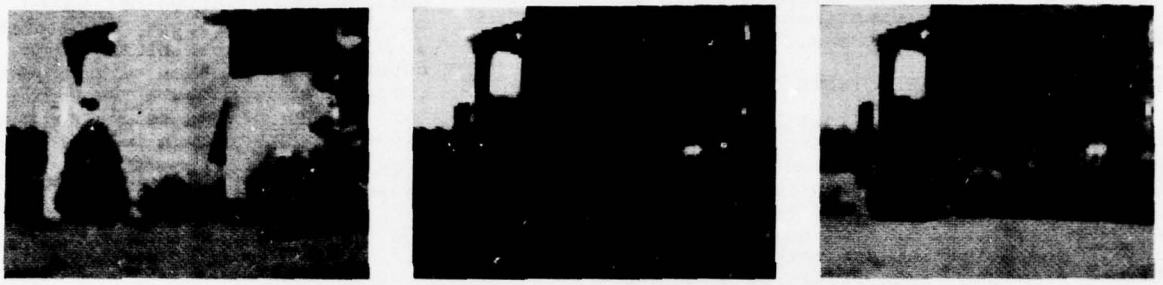


Figure 1. Red, green, and blue components of house picture.

Figure 2. Results of hand segmentation.

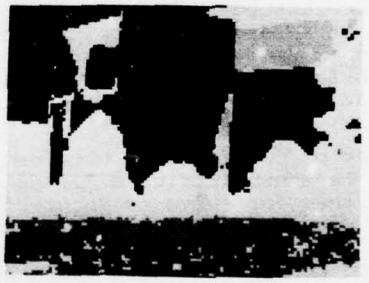


Figure 3. Results of initial classification.

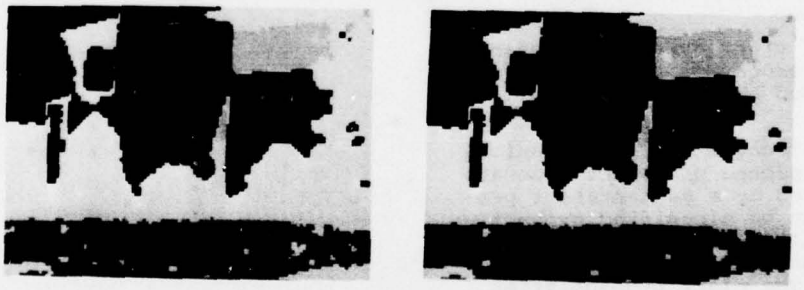


Figure 4. Results of iterations 1 and 6 of postprocessing.

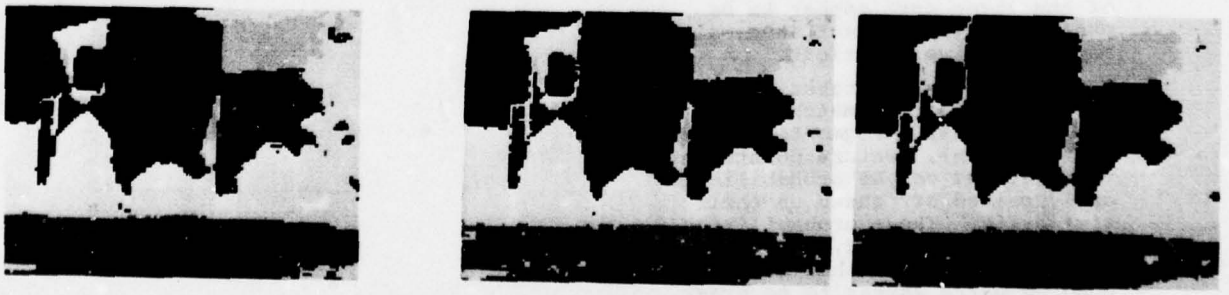


Figure 5. Results of eight iterations of relaxation.

Figure 6. Results of iterations 1 and 2 of preprocessing.

0.00	-0.01	0.00	0.00	-0.01	-0.01	-0.00	0.00	-0.01
-0.01	0.27	-1.00	-0.15	-0.01	-0.46	-0.10	-1.00	0.13
-0.02	-0.07	0.22	-0.02	0.30	-0.10	-0.22	-0.06	0.28
0.01	-0.05	-0.08	0.02	-0.12	-0.17	-0.30	0.16	-0.09
0.00	-0.10	0.17	-0.47	0.19	-1.00	-0.27	-1.00	0.05
-0.01	-0.17	-0.08	-0.25	-0.23	0.33	0.02	-0.09	-0.31
-0.02	-0.11	-0.07	0.16	0.25	0.01	0.28	0.06	0.14
-0.00	0.01	0.01	0.16	-0.25	-0.14	0.08	0.22	-0.09
0.00	-0.03	-0.25	0.10	-1.00	-0.09	-0.10	0.03	0.10

0.00	-0.05	-0.01	-0.00	-0.03	-0.01	-0.01	-0.00	-0.02
-0.07	0.53	-0.02	-0.12	-0.10	-0.34	-0.08	-1.00	0.20
-0.01	-0.16	0.39	0.04	-0.15	-0.10	0.13	-0.06	-0.02
-0.00	-0.20	0.03	0.23	0.03	0.10	-0.13	-0.02	-0.04
0.01	-0.16	0.09	-1.00	0.22	-1.00	-1.00	-1.00	-0.48
-0.01	-0.31	-0.16	-0.09	-1.00	0.32	0.15	-0.03	-0.16
-0.04	-0.03	0.02	0.14	0.53	-0.12	0.20	-0.22	0.00
-0.00	-1.00	-0.31	0.08	0.03	-0.29	0.10	0.36	-0.21
-0.01	-0.10	-0.14	0.05	-0.07	-0.09	0.02	0.05	0.36

0.00	-0.00	-0.02	-0.01	-0.03	-0.00	-0.01	0.00	0.01
-0.02	0.25	0.29	0.02	-0.20	-0.34	-1.00	-1.00	-0.02
0.00	-1.00	0.06	0.22	-0.35	-0.05	0.15	-0.01	-0.27
-0.01	-0.17	0.18	0.00	0.19	0.06	-0.03	0.06	-0.15
0.00	-0.01	-0.22	0.17	0.21	-1.00	-1.00	-0.15	-0.12
-0.02	-0.17	-0.23	-0.11	0.26	0.31	0.18	-0.39	-1.00
-0.01	-0.03	0.05	0.03	0.31	-0.33	0.08	0.16	-0.25
-0.00	-0.01	-1.00	0.15	0.07	-0.13	-0.12	0.21	0.07
-0.01	-0.10	0.20	-0.06	0.04	-0.17	0.16	-0.06	0.14

0.00	-0.01	0.01	0.00	-0.02	-0.06	-0.02	0.00	-0.04
-0.01	0.27	-1.00	0.00	-0.10	-0.27	0.02	-0.10	0.03
-0.04	-1.00	0.24	-1.00	0.26	-0.40	-0.35	-1.00	0.54
0.00	-0.05	-1.00	0.09	0.16	-0.03	-0.35	-0.26	0.10
-0.01	-0.08	-0.04	-0.07	0.34	0.15	0.05	-0.09	-0.35
-0.05	-0.53	-0.18	-0.04	0.01	0.52	0.00	-0.09	-0.18
-0.03	-0.13	-1.00	0.05	0.07	0.30	0.45	0.12	-0.06
0.00	-0.08	-1.00	-0.11	-0.04	-0.16	0.09	0.19	-0.14
0.00	-0.02	-1.00	0.13	0.12	-0.21	-0.37	-0.16	0.11

0.01	-0.11	-0.10	-0.10	-0.10	-0.10	-0.11	-0.10	-0.09
-0.11	0.62	-0.11	-1.00	-0.24	-0.35	-0.22	-1.00	-1.00
-0.10	-0.11	0.72	-0.11	-0.46	-0.30	-0.39	-0.31	-0.28
-0.10	-1.00	-0.11	0.75	-0.05	-1.00	-0.19	0.29	-1.00
-0.10	-0.24	-0.46	-0.05	0.73	-0.15	0.00	-0.15	-0.06
-0.10	-0.35	-0.30	-1.00	-0.15	0.64	-1.00	-0.48	-0.49
-0.11	-0.22	-0.39	-0.19	0.00	-1.00	0.75	-0.21	-1.00
-0.10	-1.00	-0.31	0.29	-0.15	-0.48	-0.21	0.77	-1.00
-0.09	-1.00	-0.28	-1.00	-0.06	-0.49	-1.00	-1.00	0.72

Figure 7. Mutual information coefficients obtained by applying line detectors to noise.

0.00	-0.01	-0.04	0.00	-0.01	-0.05	-0.03	0.00	0.00
-0.01	0.27	-1.00	-0.05	-0.08	-0.53	-0.13	-0.08	-0.02
0.01	-1.00	0.24	-1.00	-0.04	-0.18	-1.00	-1.00	-1.00
0.00	0.00	-1.00	0.09	-0.07	-0.04	0.05	-0.11	0.13
-0.02	-0.10	0.26	0.16	0.34	0.01	0.07	-0.04	0.12
-0.06	-0.27	-0.40	-0.03	0.15	0.53	0.30	-0.16	-0.21
-0.02	0.02	-0.35	-0.35	0.05	0.00	0.45	0.09	-0.37
0.00	-0.10	-1.00	-0.26	-0.09	-0.09	0.12	0.19	-0.16
-0.04	0.03	0.54	0.10	-0.35	-0.18	-0.06	-0.14	0.11

0.00	-0.02	0.00	-0.01	0.00	-0.02	-0.01	-0.00	-0.01
-0.01	0.22	-1.00	-0.17	-0.01	-0.17	-0.03	-0.01	-0.10
-0.02	0.29	0.06	0.18	-0.22	-0.23	0.05	-1.00	0.20
-0.01	0.02	0.22	0.00	0.17	-0.11	0.03	0.15	-0.06
-0.03	-0.20	-0.35	0.19	0.21	0.26	0.31	0.07	0.04
-0.00	-0.34	-0.05	0.06	-1.00	0.31	-0.33	-0.13	-0.17
-0.01	-1.00	0.15	-0.03	-1.00	0.18	0.08	-0.12	0.16
0.00	-1.00	-0.01	0.06	-0.15	-0.39	0.16	0.21	-0.06
0.01	-0.02	-0.27	-0.15	-0.12	-1.00	-0.25	0.07	0.14

0.00	-0.07	-0.01	-0.00	0.01	-0.01	-0.04	-0.00	-0.01
-0.05	0.51	-0.16	-0.20	-0.16	-0.31	-0.03	-1.00	-0.10
-0.01	-0.02	0.39	0.03	0.09	-0.16	0.02	-0.31	-0.14
-0.00	-0.12	0.04	0.23	-1.00	-0.09	0.14	0.08	0.05
-0.03	-0.10	-0.15	0.03	0.22	-1.00	0.53	0.03	-0.07
-0.01	-0.34	-0.10	0.10	-1.00	0.32	-0.12	-0.29	-0.09
-0.01	-0.08	0.13	-0.13	-1.00	0.15	0.20	0.10	0.02
-0.00	-1.00	-0.06	-0.02	-1.00	-0.03	-0.22	0.36	0.05
-0.02	0.20	-0.02	-0.04	-0.48	-0.16	0.00	-0.21	0.36

0.00	-0.02	-0.02	0.01	0.00	-0.01	-0.02	-0.00	0.00
-0.01	0.25	-0.07	-0.05	-0.10	-0.17	-0.11	0.01	-0.03
0.00	-1.00	0.22	-0.08	0.17	-0.08	-0.07	0.01	-0.25
0.00	-0.15	-0.02	0.02	-0.47	-0.25	0.16	0.16	0.10
-0.01	-0.01	0.30	-0.12	0.19	-0.23	0.25	-0.25	-1.00
-0.01	-0.46	-0.10	-0.17	-1.00	0.33	0.01	-0.14	-0.09
-0.00	-0.10	-0.22	-0.30	-0.27	0.02	0.28	0.08	-0.10
0.00	-1.00	-0.06	0.16	-1.00	-0.09	0.06	0.22	0.03
-0.01	0.13	0.28	-0.09	0.05	-0.31	0.14	-0.09	0.10

Figure 7, continued.

Figure 8. Results of using the coefficients of Figure 7 in a curve enhancement relaxation process.

SESSION IV

TECHNIQUES II

A SYNTACTIC APPROACH TO SHAPE RECOGNITION

K. C. You and K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

Syntactic method is used to describe the structure of a shape by grammatical rules and the local details by primitives. Four attributes are proposed to describe an open curve segment, and the angle between two consecutive curve segments is used to describe the connection. The property of the attributes and the recognition capability are studied. Two algorithms which utilize the semantic and syntactic information to perform the primitive extraction and syntax parsing at the same time are implemented. This approach attempts to develop a more general method for shape recognition.

1. INTRODUCTION

In the past years, syntactic techniques have been used for shape recognition in many applications. The general treatment of syntactic approach and a review of earlier literatures and applications can be found in the book by Fu [1]. Recently, a number of papers have reported various new results of this approach [2-10]. The syntactic method is capable of using the primitives to describe the local details and the production rules to describe the global structure. The extraction of primitives and the construction of production rules have been problems for research. If the primitives are very simple curve segments with fixed length, then we may have to use context-sensitive grammars to take care of the size problem. In fact, the complexities of primitives and production rules are flexible. We may use sophisticated production rules for simple primitives or vice versa. In this paper, we propose a set of more sophisticated primitives, so that we could hopefully solve the shape recognition problem without requiring context-sensitive grammars.

In the following paragraphs, we would refer to a shape, or a shape pattern, as the outer boundary of an object in the two-dimensional image. Conventionally, the syntactic recognition of a shape consists of three major steps. The shape pattern is first traced out from a two-dimensional image. Secondly, the shape pattern is passed through a primitive-extraction procedure, so that it can be represented in terms of primitives. Then, the representation is processed by a syntax analyzer with the knowledge of grammars. For the first step, many authors have reported the techniques for picture enhancement, boundary detection, tracing, approximation, etc. [11-13]. In this paper, we assume that the image is clear and the shape pattern can be easily extracted, since our interest is in the latter two steps.

Usually, the primitive set has a small number of elements, which are quite different from one another [1,9]. This kind of primitive set is not sufficient to describe the shapes which are similar, but slightly different in details for different classes. In our method, the primitives are defined with four attributes, which allow a large number of possible primitives. The attributed grammar [15] has a value part and a symbol part for each primitive or nonterminal. The value part may have several values called attributes. There are rules for processing the attributes corresponding to each symbolic production rule. If the attributes are considered carrying semantic information about the shape [1], we actually process both semantic and syntactic information at the same time.

The primitive-extraction procedure somehow imitates the human recognition process. In general, the human recognition of primitives is very complex, it utilizes both local and global information. Since the local information is usually used in step 2 for extraction, and the global information is used in step 3 for parsing [14], we therefore combine the two steps into one to obtain an optimal solution. That is, we use the production rules to guide the primitive extraction, or say, the extraction is embeded in the parsing.

Our approach started from the geometrical analysis of the general shape patterns, and we did not add any restriction for special applications in the development. Hopefully, the proposed method is general enough for a broader class of problems. Of course, further generalization or modification is possible.

2. ATTRIBUTED SHAPE GRAMMAR

In this section we propose a new descriptive method for curve segments and the connections of the segments. Then, the grammars used are explained.

Definition 1: A curve segment, $\overset{\frown}{X_1X_2}$, is a directional line with a starting point X_1 and an ending point X_2. The curve segment has a curvature function, $f(\ell)$, along the direction with $0 < \ell < L$, where L is the total length of the curve segment.

Definition 2: A simple curve segment is a curve segment with either $f(\ell) \geq 0$, or $f(\ell) \leq 0$, for $0 < \ell < L$.

See Figure 1 for illustrations.

To characterize a simple curve segment, we found that four attributes are sufficient. They are defined as follows.

Definition 3: The C-descriptor of a curve segment $p = \overset{\frown}{X_1X_2}$ has four attributes, $\vec{C}$, L, A, and S, i.e.

$$D(p) = (\vec{C}, L, A, S).$$

Where $\vec{C} = X_1X_2$, $L = \int_0^L d\ell$, $A = \int_0^L f(\ell)d\ell$,

and $S = \int_0^L (\int_0^s f(\ell)d\ell - \frac{A}{2})ds$.

$\vec{C}$ is the vector pointing from X_1 to X_2, L is the total length of the curve, A is the total angular change, and S measures the symmetry. Figure 2 illustrates the function of S. When p is symmetric, S = 0. If p is not symmetric, then S > 0 when p is declined to the left, and S < 0 when p is declined to the right. Somehow, S measures the degree of declination. But S measurement becomes less meaningful, when the curve segment is not simple. The four attributes do not uniquely define a curve segment unless more restrictions are added. For example, S = 0, A = 0, L = $|\vec{C}|$, the curve segment is a straight line vector, $\vec{C}$.

If a shape pattern is broken into shorter curve segments, each curve segment can be characterized by a C-descriptor, we need a primitive to describe the connections between curve segments.

Definition 4: An angle primitive is a primitive which specifies the connection between two consecutive curve segments.

Definition 5: The a-descriptor of an angle primitive, a, has only one attribute, D(a) = A, which specifies the angular change at the concatenating point of two consecutive curve segments.

Definition 6: A curve primitive is a curve segment which is not broken into shorter curve segments.

Remark: A curve primitive is not necessarily a simple curve segment.

Example 1: If a curve segment $N = \overset{\frown}{X_1X_2}$ is broken at point X_3, we may define curve segments $\overset{\frown}{X_1X_3}$, $\overset{\frown}{X_3X_2}$ correspondingly as curve primitives p_1 and p_2, with an angle primitive, a, between them. (See Figure 3.) Their descriptors are:

$$D(p_1) = (\vec{C}_1, L_1, A_1, S_1), \quad \vec{C}_1 = \overrightarrow{X_1X_3}$$

$$D(p_2) = (\vec{C}_2, L_2, A_2, S_2), \quad \vec{C}_2 = \overrightarrow{X_3X_2}$$

$$D(a) = a$$

$$D(N) = (\vec{C}_N, L_N, A_N, S_N), \quad \vec{C}_N = \overrightarrow{X_1X_2}$$

Definition 7: If a curve segment N is broken into two curve segments, N_1 and N_2, with a connection angle primitive a, then there is a production rule, $N \to N_1aN_2$.

The descriptors of them have an interesting additive property.

Theorem A: Additivity.

If $N \to N_1aN_2$ with descriptors $D(N) = (\vec{C}, L, A, S)$, $D(N_1) = (\vec{C}_1, L_1, A_1, S_1)$, $D(N_2) = (\vec{C}_2, L_2, A_2, S_2)$ and $D(a) = a$ then

$$\vec{C} = \vec{C}_1 + \vec{C}_2,\ L = L_1 + L_2,\ A = A_1 + a + A_2 \text{ and}$$

$$S = S_1 + S_2 + \frac{1}{2}\left[(A_1+a)\,L_2 - (A_2+a)\,L_1\right]$$

The proof can be found in [16].

Definition 8: $D(N_1) \underset{a}{+} D(N_2)$ denotes the above additivity.

Corollary A.1: If $N \to N_1aN_2$ then $D(N) = D(N_1) \underset{a}{\oplus} D(N_2)$.

Theorem B: The addition, $\underset{a}{\oplus}$, is associative.

If $N \to N_1a_1N_2a_2N_3$, then

$$\begin{aligned} D(N) &= D(N_1) \underset{a_1}{\oplus} D(N_2) \underset{a_2}{\oplus} D(N_3) \\ &= [D(N_1) \underset{a_1}{\oplus} D(N_2)] \underset{a_2}{\oplus} D(N_3) \\ &= D(N_1) \underset{a_1}{\oplus} [D(N_2) \underset{a_2}{\oplus} D(N_3)] \end{aligned}$$

The proof is obvious.

Because of Theorems A and B, the rules for attributes are obtained for each symbolic production rules in the attributed grammar [15]. Since a shape is a closed curve, we can define the point, which is first found in tracing as the starting and ending point. Thus, a shape is described by a curve segment with the same starting and ending point, and the angle primitive which specifies the angular change at the point. A general form of the attributed shape grammar is $G_\ell = (V_\ell, T_\ell, P_\ell, S_\ell)$ where S_ℓ is the starting symbol with special attribute ℓ, which is the label of the pattern.

$V_\ell = \{S_\ell, N's\}$

$T_\ell = \{F's, A's \mid F$: curve primitive,

A: angle primitive$\}$

P_ℓ: $S_\ell \to (XA)^*XA\{Answer_c\}$; $c \leftarrow \ell$

$N \to (XA)^*X$; $D(N) \to (D(X) \underset{A}{\oplus})^*D(X)$

where $X \in \{N's, F's\}$

{Answer$_c$} and $c \leftarrow \ell$ mean that if the parsing is successful, the shape pattern is recognized as the class labeled by c.

Since the attribute rules can be directly obtained from the production rules, they are omitted in the following example. The correspondent diagrams are shown in Figures 4 and 5.

Example 2: The shape grammar for airplane BAC 1-11, $G_c = (V_c, T_c, P_c, S_c)$

$V_c = \{S_c, N_{ci} \mid 1 \leq i \leq 8\}$

$T_c = \{F_{cj}, A_{ck} \mid 1 \leq j \leq 15, 1 \leq k \leq 7\}$

P_c:

(1) $S_c \rightarrow N_{c1}A_{c1}N_{c2}A_{c2}F_{c1}A_{c2}N_{c3}A_{c1}N_{c4}A_{c3}$

(2) $S_c \rightarrow N_{c2}A_{c2}F_{c1}A_{c2}N_{c3}A_{c1}N_{c4}A_{c3}N_{c1}A_{c1}$

(3) $S_c \rightarrow F_{c1}A_{c2}N_{c3}A_{c1}N_{c4}A_{c3}N_{c1}A_{c1}N_{c2}A_{c2}$

(4) $S_c \rightarrow N_{c3}A_{c1}N_{c4}A_{c3}N_{c1}A_{c1}N_{c2}A_{c2}F_{c1}A_{c2}$

(5) $S_c \rightarrow N_{c4}A_{c3}N_{c1}A_{c1}N_{c2}A_{c2}F_{c1}A_{c2}N_{c3}A_{c1}$

(6) $S_c \rightarrow N_{c6}A_{c2}F_{c1}A_{c2}N_{c7}A_{c4}N_{c8}A_{c3}N_{c5}A_{c4}$

(7) $S_c \rightarrow N_{c8}A_{c3}N_{c5}A_{c4}N_{c6}A_{c2}F_{c1}A_{c2}N_{c7}A_{c4}$

(8) $N_{c1} \rightarrow F_{c2}A_{c5}F_{c3}$

(9) $N_{c5} \rightarrow F_{c2}A_{c5}F_{c4}$

(10) $N_{c2} \rightarrow F_{c5}A_{c6}F_{c6}A_{c7}F_{c7}$

(11) $N_{c6} \rightarrow F_{c8}A_{c6}F_{c6}A_{c7}F_{c7}$

(12) $N_{c3} \rightarrow F_{c9}A_{c7}F_{c10}A_{c6}F_{c11}$

(13) $N_{c7} \rightarrow F_{c9}A_{c7}F_{c10}A_{c6}F_{c12}$

(14) $N_{c4} \rightarrow F_{c13}A_{c5}F_{c14}$

(15) $N_{c8} \rightarrow F_{c15}A_{c5}F_{c14}$

In Example 2, the S production rules cover the most possible starting points of the boundary. Due to the rotation of the object, the starting point may be any of the convex points. Instead of looking through the whole boundary chain for a fixed starting point, we use the S production rules to take care of these most possible starting points so that we only need to look over a short portion of the boundary chain for a sharp convex point, that would be the starting point. Because of the noise, sometimes the breaking points can not be found in extracting primitives. For instance, if the corner of angle primitive A_{c1} in Figure 4, is smeared so that F_{c5} and F_{c3} can not be extracted then we can avoid this trouble by finding A_{c4} to extract F_{c8} and F_{c4}. With this idea, the noise problem at the breaking points can be taken care of by employing different segmentation. This example has essentially two sets of segmentation, Figures 4 and 5. The non-simple curve segment, F_{c6} and F_{c10} are used as primitives to reduce the number of primitives, and consequently reduce the problem of extracting primitives from the noisy shapes. The assignment of primitives is very flexible.

3. COMPUTATION OF C-DESCRIPTORS IN DISCRETE CASE

The curvature function $f(\ell)$ in discrete case is a summation of pulse functions. It is much easier if the descriptors can be computed by additions and multiplications instead of integrations. By applying additivity, we can obtain very efficient computations. We first derive two more corollaries from Theorem A.

Corollary A.2: If N_2, in theorem A is a straight line vector, i.e. $A = 0$, $S_2 = 0$, then

$$\vec{C} = \vec{C}_1 + \vec{C}_2 \ , \quad L = L_1 + L_2 \ , \quad A = A_1 + a,$$

$$\text{and } S = S_1 + AL_2 + \frac{1}{2} A_1 L_1 - \frac{1}{2} AL$$

Corollary A.3: If N_1 is also a straight line vector, i.e. $A_1 = 0$, $S_1 = 0$, then

$$\vec{C} = \vec{C}_1 + \vec{C}_2 \ , \quad L = L_1 + L_2 \ , \quad A = a, \text{ and}$$

$$S = aL_2 - \frac{1}{2} a(L_1 + L_2) = AL_2 - \frac{1}{2} AL$$

Theorem C: Recursive Equations for Descriptors $M \triangleq (v_1 v_2 \cdots v_m)$ is a vector chain. Let M_i denote $(v_1 \cdots v_i)$, i.e. $M_m = M$, a_i is the angle between v_i and v_{i+1}, $D(v_i) = (\vec{C}_i, L_i, 0, 0)$, $i \leq i \leq m$. Then $D(M_j) = (\vec{C}_{Mj}, L_{Mj}, A_{Mj}, S_{Mj})$, $1 \leq j \leq m$, where

$$\vec{C}_{Mj} = \vec{C}_{M(j-1)} + \vec{C}_j = \sum_{i=1}^{j} \vec{C}_i \ ,$$

$$L_{Mj} = L_{M(j-1)} + L_j = \sum_{i=1}^{j} L_i$$

$$A_{Mj} = A_{M(j-1)} + a_{j-1} = \sum_{i=1}^{j-1} a_i \ ,$$

$$S_{Mj} = G_{Mj} - \frac{1}{2} A_{Mj} L_{Mj}$$

$$G_{Mj} = G_{M(j-1)} + A_{Mj} L_j + \sum_{i=1}^{j} A_{Mi} L_i = \sum_{i=1}^{j} \sum_{\ell=1}^{j-1} a_\ell L_i$$

Theorem C can be proved inductively using corollary A.2 and A.3. With this theorem, the attribute can be computed exactly instead of approximately in the discrete case. For a boundary chain of m vectors, if there is enough memory to store all the c-descriptors calculated for later processing, we need to compute $m(m+1)/2$ possible c-descriptors in the worst case. According to Theorem C, each c-descriptor needs 2 multiplications, 5 additions, and 1 shift. That implies, it takes about m^2 multiplication time to compute all the possible c-descriptors.

4. RECOGNITION OF PRIMITIVES

As mentioned previously, a curve segment can be described by four attributes. But the translation, scaling and rotation of the object may cause different values of the attributes, and also introduce different noise in digitization. Fortunately, the attributes can be transformed into a multi-dimensional space, in which the transformed descriptors are theoretically invariant under above operations if it is noise-free.

Definition 9: Transformation $T:D(p) \rightarrow T(p)$, or

$$T:(\vec{C},\ L,\ A,\ S) \rightarrow (\frac{C}{L},\ \frac{L}{L_o},\ A,\ \frac{S}{L}),$$

where $C = |\vec{C}|$, L_o is a normalization factor, which could be the total length of the shape pattern.

The proof of the invariance of the transformed c-descriptor is omitted here. Because of the invariant property, the recognition of the primitives, and hence the whole shape, are based on the transformed descriptors. If two curve segments are mirror images of each other, their transformed descriptors are only different in the sign of S/L. Consequently, if it is necessary, the storage for transformed descritors of the bisymmetric shape, e.g. top view of airplanes, can be halfly reduced by storing them in pairs. We studied the distribution of the values in the transformed space under various rotations to help us understand the noise effect on the transformed descriptors.

The details of this study and a relevant experiment can be found in [17]. For each curve primitive, the transformed descriptors under various rotations form a cluster in the multi-dimensional space. The distribution of each cluster is considerably close together and any two clusters are well separated. We also noticed that the noise at the breaking points has much bigger effect on the attributes than that at the middle of the curve segments. From above study, we suggest to recognize a curve primitive by means of a distance measure in the 4-dimensional transformed attribute space. Without losing generality, we assume each curve primitive, Q, has a referenced point, T(Q), in the 4-dimensional transformed space. If there is a curve segment, q, whose transformed c-descriptor, T(q), is considerably close to T(Q) in the 4-dimensional space, q is recognized as Q. In other words, there is a recognition function R_Q, if $R_Q(q) < t_Q$, t_Q is a threshold, $q \stackrel{r}{=} Q$. In general, R_Q can be a distance, similarity, or probability function dependent on Q. If it is also a function of Q, we could rewrite it as R(Q,q).

The recognition of the angle primitive is similar, but simpler. For an angle primitive A, there is a function R_A. If any angle a, $R_A(a) < t_A$, then $a \stackrel{r}{=} A$. If it is a function of A, it can be rewritten as R(A,a). Theoretically, the angle primitive has no length. Since sharp corners are often smoothed by noise, we allow a short length for angle primitives as called "corner tolerance". Of course, it is possible to employ the concept of partial recognition, or recognition with probability p, $0 \leq p \leq 1$, instead of "yes" and "no" for both curve and angle primitives.

5. PARSING SCHEMES

We have developed two recognition algorithms, which accept the shape pattern in form of vector chains, and perform the primitive recognition and parsing. The first one is a modified Earley's algorithm. Earley's parsing algorithm [14] consists of two parts: parsing table generation and parse extraction from the table. For classification purpose, only the first part is sufficient. Therefore, we have only modified the first part of the algorithm. The flow-chart of the modified algorithm is shown in Figure 6. The grammars used are in context-tree form as described in Section 2. $v_1 \cdots v_m$ is the unknown vector chain. T(X) denotes the transformed descriptor of a curve segment, $X \in (V_\ell \cup T_\ell) - S_\ell$, T(i,j) denotes the transformed descriptor of the subchain $v_i v_{i+1} \cdots v_j$. $I_1 \cdots I_{m+1}$ are the parse lists. For items $[A \rightarrow \alpha \cdot \beta, i]$ in I_j, $1 \leq i \leq j$, (1) iff $\alpha \neq \lambda$ (empty string), then $\alpha \stackrel{*}{\rightarrow} v_i \cdots v_j$, (2) if $\alpha = \lambda$, then $i = j$. $T(X) \approx T(k,j)$ implies that $v_k \cdots v_j$ is recognized as X, or say, $v_k \cdots v_j \stackrel{r}{=} X$. Readers may refer to [17] for details.

The Modified Earley's Algorithm

Input: A context-free shape grammar and an unknown chain of m vectors.

Output: "Accept" or "Reject"

(1) Add $[S \rightarrow \cdot\ \alpha, 1]$ to I_1 for all $S \rightarrow \alpha$ in P_ℓ

$j = 1$

(2) (a) If $[N \rightarrow \alpha \cdot B\beta, i]$ is in I_j and $B \rightarrow \gamma$ in P_ℓ
then add $[B \rightarrow \cdot\ \gamma, j]$ to I_j

(b) If $[N \rightarrow \alpha \cdot\ , i]$ is in I_j
then for all $[B \rightarrow \beta \cdot N\ \gamma, k]$ in I_j
add $[B \rightarrow \beta N \cdot \gamma, k]$ to I_j

(3) $j = j+1$
if $j > n+1$ goto (4)
For all $[N \rightarrow \alpha \cdot X\beta, i]$ in I_k, $1 \leq k \leq j$
$X \in \{F's, A's\}$

(a) If $\beta \neq \lambda$ and $T(X) \approx T(k,j)$
then add $[N \rightarrow \alpha X \cdot \beta, i]$ to I_j

(b) If $\beta = \lambda$, $T(X) \approx T(k,j)$ and $T(N) \approx T(j,i)$
then add $[N \rightarrow \alpha X \cdot, i]$ to I_j

(4) If $[S \rightarrow \alpha \cdot\ , 1]$ in I_{m+1} for some α,
then "Accept", otherwise "Reject"

In some recognition problems, only finite state grammars are used. Therefore, we also developed a finite automaton which embeds the primitive extraction. Since we always can find an angle primitive following a curve primitive, we consider that each time the input contains a curve primitive and an angle primitive. Figure 7 shows the storage of a finite state grammar in a structural form. The recognition, with its flow-chart shown in Figure 8, uses a STACK. Each element in the STACK contains two fields, state and vtpt, a state and a vector

pointer, which means the first vtpt-1 vectors of the unknown shape have been accepted through the state. FS is a set of final states.

The Finite State Automaton

Input: A finite-state shape grammar in tabular form (Figure 7) and an unknown chain of m vectors.

Output: "Accept" or "Reject"

Method:

(1) kp ← 1
    STACK (kp) ← (S_0,1)

(2) If kp = 0 then terminates with "Reject"
    otherwise s ← state (STACK (kp))
              p ← vtpt (STACK (kp))
             kp ← kp - 1
             tp ← PTR (s)

(3) If tp = 0 GOTO (2)

(4) nxp ← nxpt (TABLE (tp))
      F ← curve (TABLE (tp))
      A ← angle (TABLE (tp))
    nxs ← nxst (TABLE (tp))

(5) For all x,y, p ≤ x < y ≤ m+1
    If (T(p,x) = T(F) and T(x,y) = T(A)) then
       if nxs ε FS and y = m+1 then GOTO (7)
       otherwise kp ← kp+1, STACK (kp) ← (nxs,y)

(6) If nxp = 0 then GOTO (2)
       otherwise tp ← nxp, GOTO (4)

(7) Terminates with "Accept"

The modified Earley's algorithm basically implements a breadth-first search, while the automaton implements a depth-first search. They both search for feasible primitives which satisfy the production rules. The automaton recognizes the primitives, while the Earley's algorithm recognizes the nonterminals as well as the primitives. If we abandon the recognition of nonterminals in the Earley's algorithm, the two algorithms will end up with the same classification result. But, the automaton would be faster, because it stops at the first feasible set of primitives found. The recognition of nonterminals upgrades the discriminating power of the Earley's algorithm.

Figure 9 has 3 views of 2 airplane models. V, U, T indicate different angle views, they are all close to the top view. (a) is different from (b), (c) by two small missile-tails and a machine gun on the right wing. But the machine gun may not appear in the digital picture. We can construct a grammar, $G_{F86,T}$, to distinguish it from MIG-15. $G_{F86,T}$ can be in context-free form or in finite-state form, and both algorithms are applicable. If we want to distinguish the two views of MIG-15, we can construct a grammar $G_{MIG,U}$ to distinguish them. Since the major difference between the two views is the width of the fuselage closed to the tail and the whole tail is not designed as a primitive, we need to check the nonterminal which represents the whole tail or the whole shape excluding the tail. In such a case, the Earley's algorithm can discriminate better than the automaton.

If a partial recognition or probabilistic recognition is used for primitives, the above two algorithms can be further modified to exhaust all the cases and select the best acceptable one, or the most probable one, among all the classes. In such a case, the finite automaton has no advantage over the Earley's parser. The computational cost of the parsing algorithms increases rapidly with n, where n is the number of vectors in the boundary chain. Therefore, we try to smooth the boundary and reduce the number of vectors before parsing if the smoothing does not distort the boundary very much.

Both of our parsing algorithms have been implemented on a DCD 6500 computer in FORTRAN language. They are used to classify the airplane shapes in Figures 9 and 10. It took about 0.35 secnd per grammar for a boundary chain of 60 vectors.

References

[1] Fu, K. S., Syntactic Methods in Pattern Recognition, Academic Press, 1974.

[2] Fu, K. S. (ed.), Syntactic Patter Recognition Applications, Springer-Verlag, 1977.

[3] Stockham, G. C., L. N. Kanal, and M. C. Kyle, "Design of a Waveform Parsing System," Proc. of First Int'l Joint Conf. on Pattern Recognition, (Oct. 1973), pp. 236-243.

[4] Feng, H. T. and T. Pavlidis, "Decomposition of Polygons into Simpler Components: Feature Extraction for Syntactic Pattern Recognition," IEEE Trans. on Computers, Vol. C-24, (June 1975), pp. 636-650.

[5] Moayer, B. and K. S. Fu, "A Syntactic Approach to Fingerprint Pattern Recognition," Pattern Recognition, Vol. 7, (1975), pp. 1-23.

[6] Moayer, B. and K. S. Fu, "A Tree System Approach for Fingerprint Pattern Recognition," IEEE Trans. on Computers, Vol. C-25, (1976), pp. 262-274.

[7] Pavlidis, T., "Syntactic Feature Extraction for Shape Recognition," Proc. of Third Int'l Joint Conf. of Pattern Recognition, Coronado, CA, (Nov. 1976), pp. 95-99.

[8] Pavlidis, T., "Syntactic Pattern Recognition on the Basis of Functional Approximation," in Pattern Recognition and Artificial Intelligence (C. H. Chen, Ed.), Academic Press, 1976, pp. 389-398.

[9] Pavlidis, T. and F. Ali, "A General Syntactic Shape Analyzer," Tech. Rep. No. 221, Computer Science Lab., Princeton University, December 1976.

[10] Pavlidis, T., "A Review of Algorithms for Shape Analysis," Tech. Rep. No. 218, Computer Science Lab., Princeton University, September 1976.

[11] Pavlidis, T. and S. L. Horowitz, "Segmentation of Plane Curves," IEEE Trans. on Computers, Vol. C-23, (Aug. 1974), pp. 860-870.

[12] Sidhu, G. S. and R. T. Bonte, "Property Encoding: Application in Binary Picture Encoding and Boundary Following," IEEE Trans. on Computers, Vol. C-21, No. 11, Nov. 1972, pp. 1206-1216.

[13] Rosenfeld, A., "Survey: Picture Processing 1975," Computer Graphics and Image Processing 5, (1976), pp. 215-237.

[14] Aho, A. V. and J. D. Ullman, The Theory of Parsing, Translation and Compiling, Vol. 1: Parsing, Prentice-Hall, 1972.

[15] Lewis, P. M., D. J. Rosenkrantz, and R. E. Stearns, Compiler Design Theory, 1972.

[16] You, K. C. and K. S. Fu, "Syntactic Shape Recognition," Image Understanding and Information Extraction, Quarterly Report of Research, Nov. 1, 1976 - Jan. 31, 1977, (T. S. Huang and K. S. Fu), TR-EE 77-16, (March 1977), Purdue University, pp. 72-83.

[17] Fu, K. S. and K. C. You, "Syntactic Shape Recognition," Image Understanding and Information Extraction, Semiannual Summary Report of Research, April 1, 1977 - Sept. 30, 1977, (T. S. Huang and K. S. Fu), TR-EE 77-41, (Nov. 1977), Purdue University, pp. 52-64.

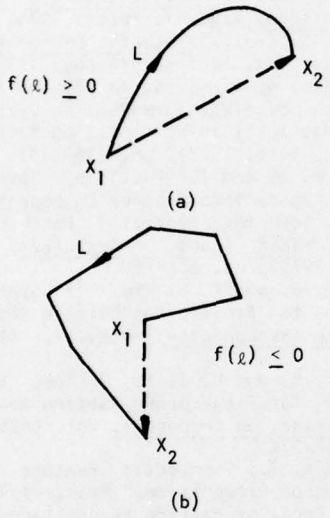


Figure 1

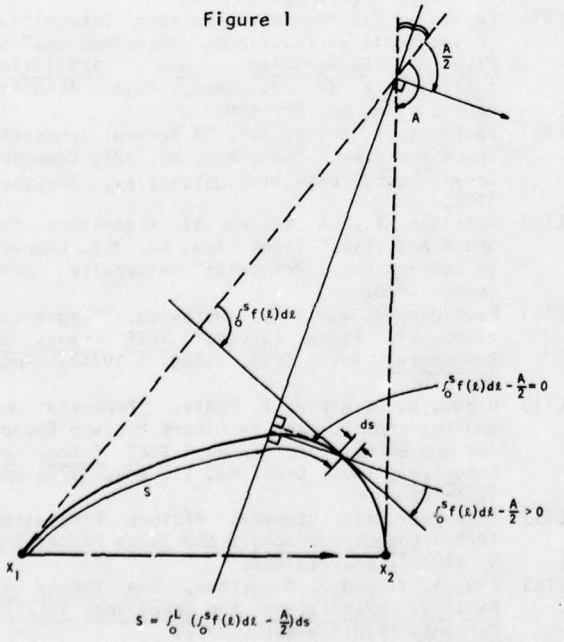


Figure 2

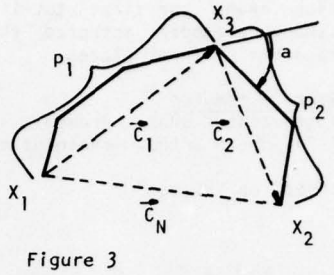


Figure 3

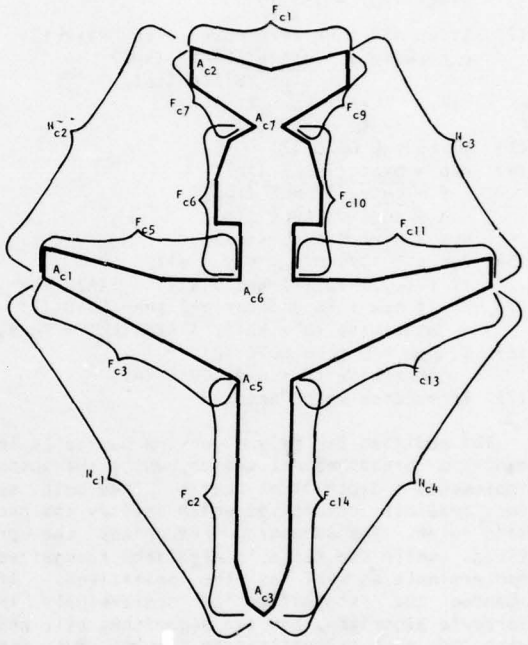


Figure 4

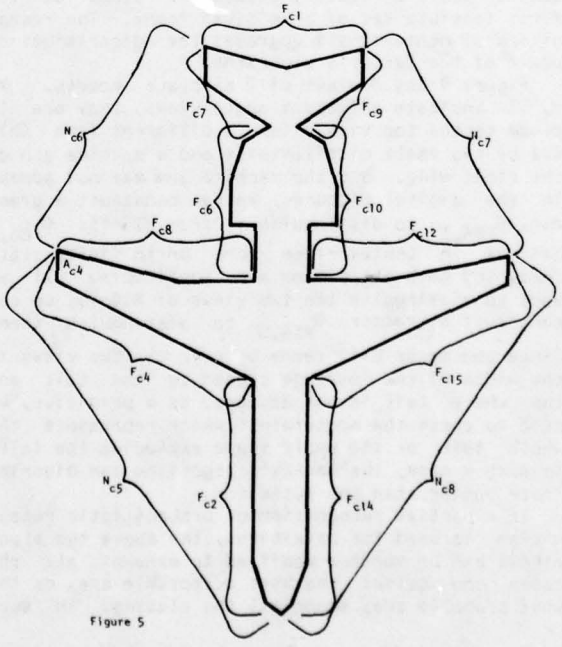


Figure 5

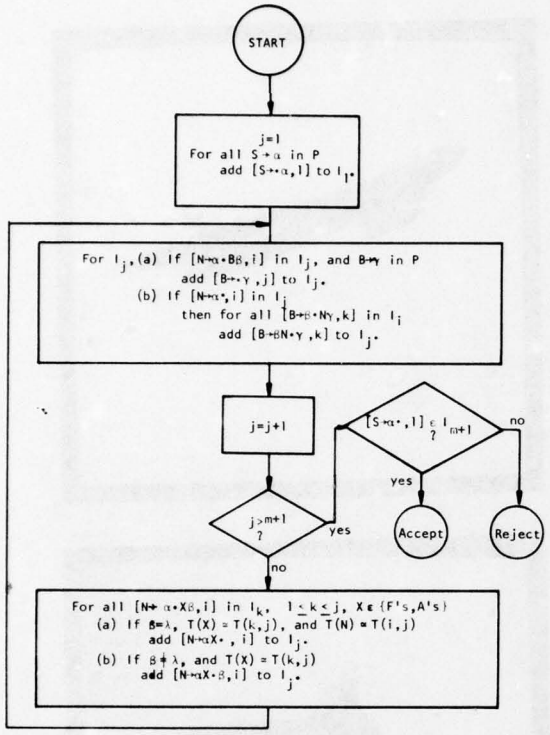


Figure 6

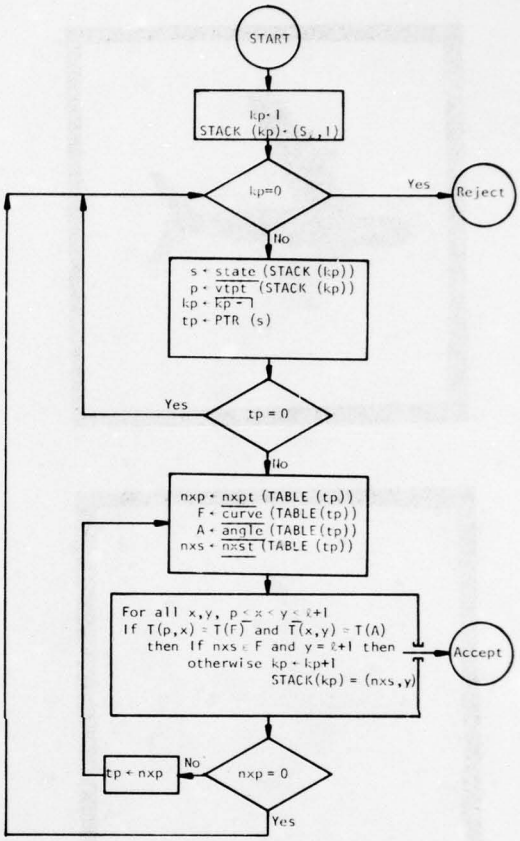
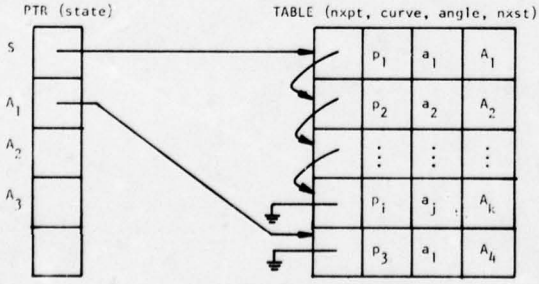


Figure 8

$S \cdot p_1\ a_1\ A_1 \quad p_2\ a_2\ A_2 \quad \cdots \quad p_i\ a_j\ A_k$

$A_1 \cdot p_3\ a_1\ A_4$

(a) Production Rules



(b) The structural form of Finite State Shape Grammar

Figure 7

Figure 9

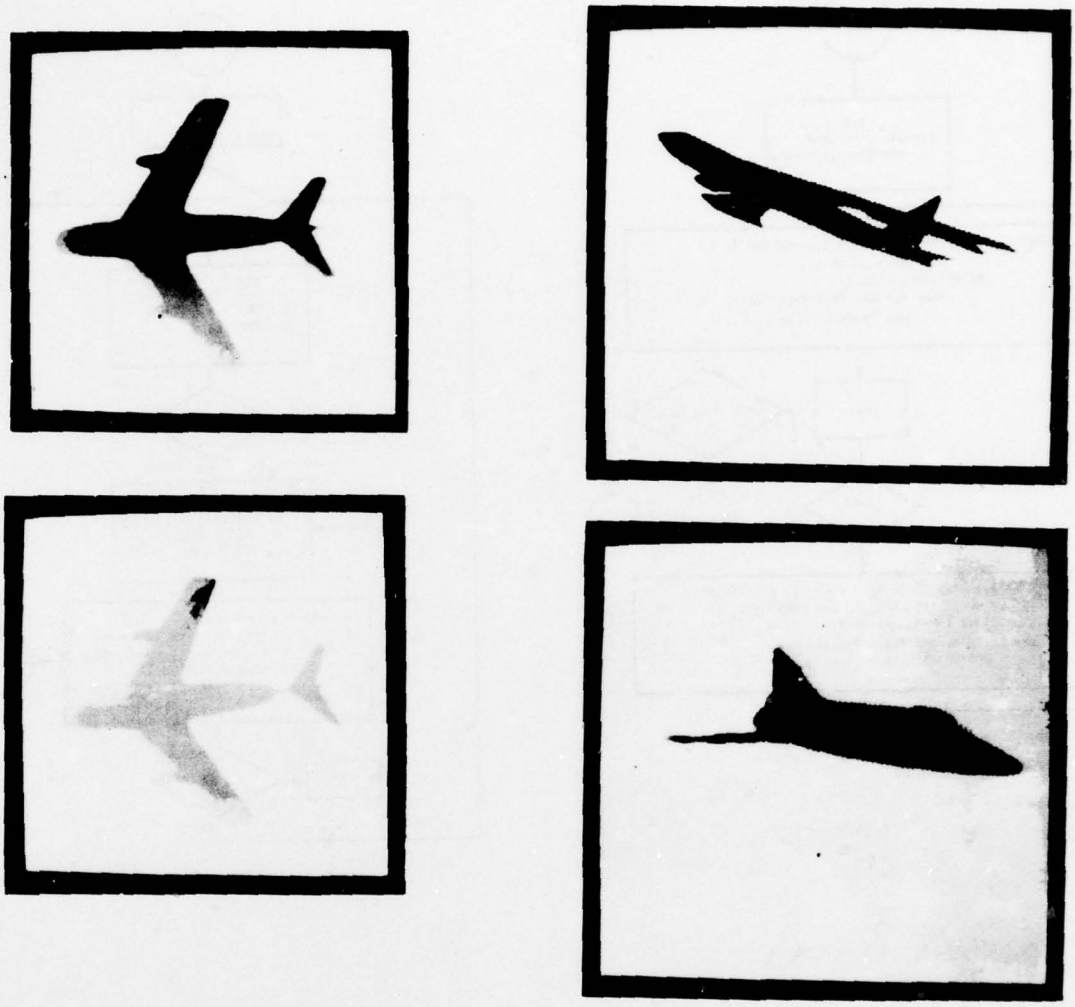


Figure 10

A POSTERIORI IMAGE RESTORATION

John B. Morton
Harry C. Andrews

Image Processing Institute
University of Southern California
Los Angeles, California 90007

Two algorithms are developed which address the problem of estimating the magnitude and phase of the optical transfer function associated with a blurred image. The primary focus of the research is on the estimate of the phase of the optical transfer function. Once an estimate of the optical transfer function has been made, the corresponding blurred image is Wiener filtered to estimate the original unblurred image (the object). Results are demonstrated on computer simulated blurs and also on real world blurred imagery.

The technique to be studied attempts to remove degradations from an image using a minimum of knowledge. The following assumptions will be made:

a) a blurred image is available
b) the PSF is spatially invariant
c) the extent of the PSF is small compared to the extent of the image.
d) the blurred image is relatively noise free (i.e. the dominant degradation is blur and not noise).

The emphasis will be on estimating the complex OTF; that is both magnitude and phase of the OTF. Once the OTF has been estimated, techniques known to be successful given knowledge of the OTF will be used to estimate the undegraded image.

The general philosophy will be to assume all quantities are continuous, and any discretizations are a corruption of the continuous process and introduce errors into the system. For example the image, $f(x,y)$, is assumed to represent a continuous function. Since convolutions of continuous functions are continuous functions, the blurred image, $g(x,y)$, will also be assumed to be continuous.

Dividing the degraded image into subimages, which may overlap, and indexing the subimages by i,

$$G_i(u,v) \cong H(u,v)F_i(u,v)$$

or equivalently

$$G_i(u,v) = H(u,v)F_i(u,v)+E$$

where E is the error inherent in the above approximation. Forming the product we obtain

$$G_i(u,v)G_i^*(u+\Delta u,v+\Delta v)\cong H(u,v)H^*(u+\Delta u,v+\Delta v) \cdot F_i(u,v)F_i^*(u+\Delta u,v+\Delta v)$$

If the product $H(u,v)H^*(u+\Delta u,v+\Delta v)$ can be estimated and given that $H(0,0) = 1$, we obtain a recursive relationship where

$$H^*(u+\Delta u,v+\Delta v) = \frac{\frac{1}{N}\sum_{i=1}^{N} G_i(u,v)G_i^*(u+\Delta u,v+\Delta v)}{H(u,v)\frac{1}{N}\sum_{i=1}^{N} F_i(u,v)F_i^*(u+\Delta u,v+\Delta v)}$$

Now considering the phases and observing that $\theta_H(0,0)=0$, we have a recursive algorithm

$$\theta_H(u+\Delta u,v+\Delta v) = \theta_H(u,v) - \left(\overline{\theta_{G_i}(u,v)-\theta_{G_i}(u+\Delta u,v+\Delta v)}\right) + \left(\overline{\theta_{F_i}(u,v)-\theta_{F_i}(u+\Delta u,v+\Delta v)}\right)$$

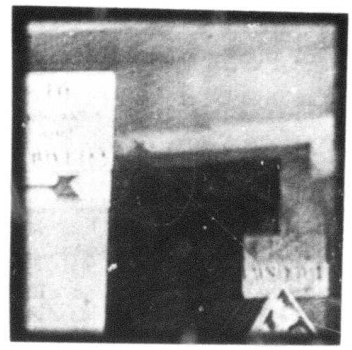
where the bar element denotes averaging in some sense.

Techniques have been developed, based upon the above equations to estimate the complete complex OTF from a blurred image. Results on real world arbitrary blurs are presented in figure 1. Here two original scenes are partitioned into subregions, OTF's calculated, and then Wiener filter restored.

a) Original

b) Restored

c) Original



d) Restored

Figure 1. A Posteriori Blind Restoration

Target/Background Segmentation and Classification in FLIR Imagery

O. R. Mitchell

Purdue University
West Lafayette, Indiana 47907

Abstract: Ongoing projects at Purdue are producing results in the areas of real-time target tracking and classification. This paper presents some techniques which are being used on digitized images and their utility for segmentation and classification of targets in FLIR imagery. The segmentation algorithm assumes potential target objects are already located in the image and the required operation is to precisely separate the object from its background. To accomplish segmentation, background features are measured over a region surrounding but not including the object region. Then the features of the object region are measured and compared to those of the background. Pixels not matching the background are labeled as object points. The features measured are grey level, edges, and texture. A method of classification of the segmented objects using projections is next presented and discussed.

I. Segmentation: We have been collaborating with Honeywell System and Research Division in developing a system to detect and recognize tactical targets in FLIR (forward looking infrared) imagery. Shown in Fig. 1 are 16 sample FLIR tactical targets. These images are thermal and several characteristics apply to active vehicles: (1) the motor is sometimes visible as a hot (bright) spot, (2) edges can be detected on the object/background boundary, and (3) the average temperature (grey level) of the object is often different from the background. These characteristics are presently used by a Honeywell Autoscreener System to locate potential target areas. The images in Fig. 1 were selected by the Autoscreener as potential targets. Note that four of these are false alarms.

The techniques described here assume that the location of each potential target is known. They attempt to separate the target and nontarget points based on features measured in the background and in the target region.

A. Segmentation Features: To accomplish target segmentation, background features are collected over an annular region surrounding the potential target. Then the features of the target region are compared to those of the background, the points not matching the background are labeled as target points. As is evident from the sample images, grey level alone is not always enough information for accurate segmentation,

Fig. 1 Original FLIR imagery selected by initial processing as potential targets. Numbered left-to-right, top-to-bottom are: armored personnel carrier (APC) - 1, 4, 8, 11; truck - 2, 3, 6, 13; tank - 5, 9, 12, 15; false alarm - 7, 10, 14, 16. Each image is 128x128 with 6 bits of grey level.

so additional features are necessary for the process. Hopefully, once the right features are selected, any points that have different features than the surrounding background points will be part of the target.

The two additional features are chosen to complement the grey level image. These are texture and edges. The texture was chosen because it seems probable that object and background textures would not be identical, assuming a good texture measure were available to differentiate among textures. The edges were chosen as a feature due to the predominance of edges along the target background interface and the fact that grey level (temperature) and texture become ambiguous near the object boundaries.

The edge feature is a gradient type measurement measured over a 7x7 window for each point. The absolute difference between the upper 21 points and the lower 21 points is compared against the absolute difference between the left 21 points and the right 21 points. The center point is then replaced by the maximum of the absolute values of these two differences. This process is repeated for each point in the original image to produce the edge feature image. Fig. 2 contains edge feature images based on the original images in Fig. 1.

Fig. 2 Edge features images as measured on original images in Fig. 1.

The texture feature is derived from the max-min local extrema described elsewhere [1-2]. Local grey level extrema are measured in hysteresis smoothed versions of the original image using three smoothing thresholds. The lowest level extrema correspond mostly to noise in the image, whereas the highest correspond mostly to edges. The remaining medium level extrema are a measure primarily of the texture in the image. The medium level extrema locations for the targets in Fig. 1 are shown in Fig. 3.

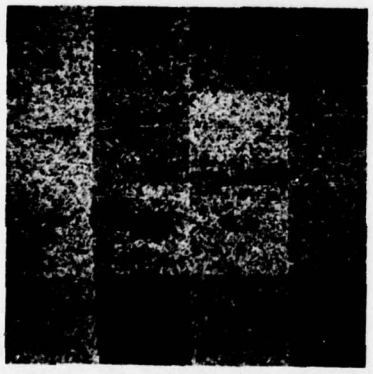


Fig. 3 Medium level extrema Locations for the targets in Fig. 1.

The texture feature image is created from the extrema by averaging the number of medium level extrema in every 10x10 window in the image and replacing the center point of the window with the average. Texture feature images are shown in Fig. 4.

B. Segmentation Procedures: Once the feature images are produced two concentric circles are centered at each potential target as derived from the Honeywell preprocessing system. The inner circle represents the potential target area and the annular region between the two circles represents the background region. In an automated system these circle sizes would be adaptive since approximate target size and background context will also be available from prior processing stages. In our implementation of the system here, the inner radius was fixed at 40 pixels and the outer radius at 64 pixels.

Fig. 4 Texture feature images derived by averaging the number of extrema over 10x10 windows.

The background annular region must be large enough to allow a sufficient background sample to be collected but it must not include target points or be so large that irrelevant background obscures the background/target differences.

The present background statistics gathering program generates a three-dimensional histogram over the original and two features for all background points. The quantization selected allows for 32 original grey levels, 8 edge values, and 16 texture values. This background histogram is therefore composed of 4096 bins. Once the background 3-D histogram is completed, each potential target point (3-D vector) is compared against its background bin. If that feature combination occurs often in the background, the point is considered another background point. If the feature combination does not occur in the background, that point is labeled a target point.

The target test was done over the whole image instead of just the inner circle to give us an idea of the background rejection of our process. Segmentations using this process are shown in Fig. 5.

Fig. 5 Segmentation results on the originals in Fig. 1 using grey level, edges, and texture. The detected target points are left at their original grey level and the detected background points are turned off.

II. Classification By Projections: The segmentations produced by the method previously described produce results which are sometimes fragmented and contain drop-out and extraneous points. A classification scheme which is somewhat insensitive to these variations would be appropriate. We are presently investigating the use of projections through the segmented object to derive classification features. A similar type of structure recognition method is being developed by New Mexico State University for missile tracking at the White Sands Missile Range [3]. It has the advantage that the integration process of the projections averages out many of the noise problems inherent in thermal images and our segmentation method.

Shown in Fig. 6 are eight projections through a segmented object (background points set to zero, target points remain that their original grey level). the object is the APC which is the 11th image in Fig. 5. The small circles along the horizontal axes represent 10% area increments along the projections. The numbers printed below are the distances between the 10% area increments normalized so that the total distance (representing 65 pixels horizontally) is 1000. The narrowest and widest projections are then selected by measuring the distance occupied by the center 60% of the area. This approximately removes the rotation dependence of the projections. Classification is made on the remaining two projections.

Fig. 6 Eight projections through the eleventh segmented target (APC) in Fig. 6. The small circles below the horizontal axes are 10% area increments. The numbers indicate normalized distances between circles.

Fig. 7 includes the narrowest and widest projections for one sample of each type target. Distinguishing characteristics of the projections for these particular categories are:

(1) APC - A significant dip in the center of the wide projection representing the seating area; the temperature (brightness) on either side of the dip is comparable; the dip often shows in the narrow projection as well.

(2) Truck - A bright body area, a small dip representing the windshield, and a smaller region representing the front, all visible in the wide projection.

(3) Tank - A predominant bright motor in both projections and a dip in the wide direction due to vent holes.

(4) False alarm - Highly varing projections due to disjoint points; non-symmetrical narrow projection; often a small total number of points.

These characteristics can best be measured by the location and size of local extrema along the narrow and wide projections.

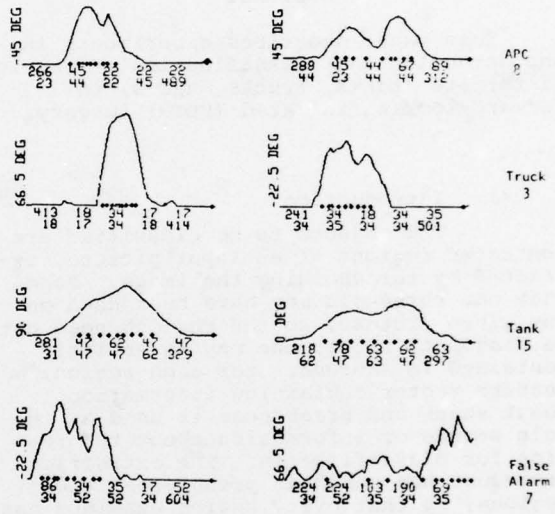


Fig. 7 Narrowest (left) and widest (right) projections through four sample objects. Labels for each pair of projections indicate the type of target and its numerical position in Fig. 5.

References:

1. O. R. Mitchell, C. R. Meyers, and W. A. Boyne, "A Max-Min Measure for Image Texture Analysis," IEEE Transactions on Computer, Vol. C-26, April 1977, pp. 408-414.

2. S. G. Carlton and O. R. Mitchell, "Image Segmentation Using Textures and Grey Level," Proceedings of the IEEE Conference on Image Processing and Pattern Recognition, Troy, N.Y., June 6-8, 1977, pp. 387-391.

3. G. M. Flachs, W. E. Thompson, and Yee-Hsuun U, "A Real-Time Structural Tracking Algorithm," NAECON 1976 Record, pp. 161-168.

RESULTS IN FLIR TARGET DETECTION AND CLASSIFICATION

D. L. Milgram

Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

This paper describes experiments in the detection and classification of tactical targets (tanks, trucks, APC's) in forward-looking infrared (FLIR) imagery.

1. Introduction

The objects to be classified are connected regions of an input picture, extracted by thresholding the image. More than one threshold may have been used on any given picture, so the regions need not be disjoint; rather, one may be entirely contained in another. For each region, a feature vector containing information about shape and brightness is used as the sole source of information about the region for classification. The extraction procedure has somewhat preselected these regions, so that every region examined has at least minimal correspondence (20%) between its perimeter and the high-edge points, has at least minimal contrast (.2 gray level), and is of roughly appropriate size (between 20 and 1000 pixels). For a description of the Superslice region extraction process, see [1-2].

2. Stage 1: Preclassification

The classification can be thought of as a two-stage process (shown schematically as Figure 1). The first stage is a crude "semantic" classifier which identifies some regions as having properties which indicate that they are not targets. Thus, all targets have relatively similar height and width, seen at any aspect angle. Any region with h/w greater than 3 or less than 1/3, then, may be confidently rejected from further consideration. Similarly, targets "should" show some minimal contrast at their perimeters, a good edge-perimeter overlap, and small targets should be of nearly uniform brightness. All these criteria are set by establishing numerical thresholds such that at least 95% of the sample targets satisfy the criteria.

This is called "semantic" classification, rather than a very crude statistical classification, because the particular criteria used have been chosen to distinguish the targets on the basis of physical characteristics of true target images. A statistical classifier, even if it arrived at the same scheme, would be assessing discriminatory ability on the sample of classified regions provided for training, and could reflect any peculiarities which happened to distinguish the categories in that sample. (In the NVL data, APC's often exhibit an asymmetry which is due to the fact that most of those in the sample appear in only a single aspect. An apparently good statistical classifier could be formed which would unhesitatingly identify any APC in some other aspect as a tank.)

This pre-classification examines individual features to determine whether they could be reasonably associated with true targets, and discards "ridiculous" cases. A side-effect of this sorting is to assure that feature values seen by the subsequent statistical classifier are never very far from their characteristic values. This makes the classifier much better-behaved than one which accepts non-normally distributed features (as most do) that have not been "critiqued."

3. Stage 2: Statistical Classification

Once the set of extracted regions has been reduced to a set of bright, compact, reasonably uniform regions, statistical classification is used to assign a class to each particular combination of features (or rather, to its associated region). A great many kinds of statistical decision rules exist. Access to the MIPACS [3] interactive system allowed us to design a decision tree (each node of which is a standard classifier) for efficient classification. The system allows individual decision functions to be either linear (e.g., Fisher), quadratic, or maximum likelihood, and provided a convenient mechanism for selecting which decisions to make, and just which features to use at each decision point.

The basic structure selected is

shown in Figure 2. The first node actually represents a non-statistical selection. Because of the wide range of apparent sizes of the target images (from 25 to 1000 pixels) and the consequent wide range in visible complexity of detail, it was quickly determined that statistical classifiers would not provide good discrimination over the entire size range. (Almost every feature measured showed substantial correlation with apparent size, and since the various sample classes happened to have rather different image size distributions, our earliest classifiers used that factor as a main classification indicator.) Therefore the first step in the classification is a simple split on image area -- with all regions of less than 95 pixels going to the "small" subtree, and the remainder passing into the "large" subtrees. For several reasons, principally a presumed lesser urgency for detailed identification of small or distant objects and the fact that in the smallest images no significant differences between the various target classes are apparent, the small regions are simply sent to a node which classifies them as (small) "target" or "non-target" -- the specific type of target is left unspecified. For the large regions, a two-stage process followed. As neither APC's nor trucks are particularly well characterized by the features used and their distributions are very similar, they were merged into a composite "truck-like" class. Any region found to be in this class is then assigned as APC or truck by a Fisher discriminant. (A major reason for this breakdown is that it permits fairly large samples to be used at an important decision point and relegates use of the sparsely sampled truck class to a relatively inconsequential discrimination.) The principal decision was therefore between the "tank" and "truck-like" classes and the "non-target" class. Our approach applied the maximum likelihood criteria directly to the tank, truck-like, and non-target classes.

Given the true structure for the classification, the kind of classifier and the set of features at each node were determined. The number of features which can reliably be used depends on the size of the sample set used for training. Assuming that the features are chosen so as to avoid apparent vagaries in the set of exemplars, one can confidently use one feature for each ten samples in the smallest group up to a limit of one-third the sample number for a linear classifier. As quadratic classifiers utilize more detail of the presumed distribution one is restricted to the conservative end of that range. These rules of thumb, while not universally valid, are nonetheless useful guides.

By merging the truck and APC classes, we allow comfortable use of a quadratic classifier on five or six features at the main decision node, while the smaller samples make a linear classifier or a three (or four) feature quadratic classifier more reasonable at the lower node. The "small" node could utilize five or six features -- but one is hard-pressed to find even that many which provide any discriminatory power at all.

4. Experimental Results

4.1 Feature selection

As in any classification problem, much of the initial feature selection for the vehicle recognition task was carried out informally. This phase is largely introspective, determining characteristics of the images that seem helpful for human judgment, then identifying some features that should suitably reflect these characteristics. This initial feature set (conveying "shape" and "relative brightness") is listed in Table 1. All of these features seem appropriate for use with linear or quadratic classifiers.

The features were examined in several ways. First, histograms for each feature were produced for every sample class. These histograms were examined to see whether the sample distributions satisfied the criteria noted in the last section. The differentiation that appeared was interpreted as to whether it was a true difference between classes, or simply a sampling anomaly. (At this stage too, particular features might be replaced by similar features of slightly different functional form, to better satisfy the requirements of automatic classification.) Second, those features that seemed to have some merit were ranked for classification power at each node of the decision tree. The "Automask" method, available within MIPACS, was used. Briefly, Automask finds, for each feature, its "share" of the total dispersion both between and within sets, and finds the single feature which produced the greatest comparative variance between sets. This feature is then deleted from consideration, and the other features reexamined to find the next best feature, and so on. The relative merits of the features for each node are shown below.

Node	Good features	Usable features
Small	E&P	(h/w)', (h*w)/A, (h+w)/P, diff, skewness, asymmetry
Large	E&P, diff	(h/w)', (h*w)/A, skewness, asymmetry, E_p

Trucklike	E_p, asymmetry	(h/w)', (h+w)/P, skewness, E&P

Shape features:

In the first stage, the (h/w)' height-to-width feature was useful in identifying small bright streaks as non-targets. In the statistical classifier for small targets, shape features were individually very weak in distinguishing targets from non-targets. For large targets, diff was the best shape feature at node LARGE; all others but asymmetry were also of some use. At node TRUCK-LIKE, on the other hand, asymmetry was the best shape feature, with the remainder of no value.

Brightness-related features:

Edge-border coincidence (E&P) was by far the strongest single feature for both nodes involving target/non-target discrimination (OBJ and LARGE). For small targets, it provides nearly all the discrimination in the second stage. For large targets, it provides evidence which is well complemented by shape information -- both must be included for adequate performance. Also very useful, particularly at stage 1, is E_p, which provides substantially different information from E&P. Gray level variance is used to some effect in the first classifier stage, but is not effective in the second stage. Perimeter contrast information appears to be much more effectively conveyed through E_p than dgl.

These rankings, while not dependable when taken alone, have been very helpful in suggesting which features could usefully be included in decisions at each node and which should be omitted. This was especially helpful in the case of the shape features, for which estimates of relative merit were not obtainable.

The final stage of feature testing was experimental. Features suggested either by Automask or by the problem definition were included in decision functions, and self-classification attempted. In many cases, the results were not satisfactory and one or more features were added or deleted until "good" results were obtained. If too many features were present in this classifier, features were removed until the best classification obtained with an acceptable number of features was found.

4.2 Classification

The NVL data base as windowed for classification purposes consists of:

75	Tanks
34	Trucks
55	APC's
164	Target windows
10	Non-target windows
174	Total windows

Associated with each window was a liberal threshold range extending from the shoulder of the background peak gray level to the highest gray level at which there was significant sensor response. Although these ranges were manually selected, this is not a significant interference with the automatic nature of the algorithm since the gray level ranges can be chosen by a simple scheme which identifies the background peak and proposes every threshold above the peak. (If a coarse temperature calibration is available, this task is even simpler.)

The Superslice algorithm was run on these windows using the selected gray level ranges. Connected components whose contrast, edge-perimeter match score and size were within tolerance were retained. The resulting sets of regions of a window may be described by containment forests. Within each containment tree, Superslice selects for the candidate object region its best exemplar(s) based on edge match. Thus, every tree has one or more best exemplars associated with it.

Each containment tree is manually labelled as either "target-related" (containing regions associated with the target) or noise (spatially apart from a target region) so that false dismissals can be determined.

Of the 164 target windows, two windows had containment forests with no target-related regions present. At this stage, the false dismissal rate is 2/164 or 1% for Superslice. Determination of a false alarm rate is inappropriate since the discrimination performed by Superslice is "object vs. non-object", not "target vs. non-target", and there is no ground truth for the number of objects (including targets, hot rocks, trees, etc.) in the frames.

The next stage - preclassification - performs possible-target vs. non-target screening. [For the purpose of building the screening criteria and subsequent classifier, a single exemplar per target was hand-chosen. All noise regions, however, were retained.] Of the 162 target windows, the preclassifier retained 161 for a false dismissal rate of 1%. In addition, 44 noise exemplars also survived as possible targets. The false dismissal was small and very faint.

After preclassification, 150

selected target exemplars and all noise exemplars were split into a training set (74 targets and 22 noise regions) and a test set (76 targets and 22 noise regions). The training set was used to design the optimum decision rule. It was felt that similar results in classifying both sets would then indicate that the classifier had utilized robust characteristics of the target class and thus could be expected to give similar results on further data of the same type.

A linear discriminant is used at two nodes: for the small target/non-target and for the truck/APC discriminations. Five features were used at both nodes, of which four were the same: $(h*w)/A$, $(h+w)/P$, asymmetry, E&P. The fifth feature was diff, for the small target discriminant, and skewness, for the truck/APC discriminant. The large targets are divided into three classes (tank, truck/APC, other) by a quadratic maximum likelihood discriminant using six features: $(h/w)'$, $(h*w)/A$, diff, skewness, E&P and E_p.

The detection results of the fixed class classifier on the 150 selected target exemplars are summarized by:

	Train	Test	Total
Large	53/53	53/55	106/108
Small	20/21	20/21	40/42
Total	73/74	73/76	146/150

where "M/N" means "M successes out of N tries." The classifier thus appeared to be robust.

We say that a false dismissal for a window containing a target has occurred when no target exemplar (at any of the thresholds) is classified as a target (i.e., classified as tank, truck or APC). Similarly, a false alarm is any noise exemplar (i.e., not associated spatially with a target region) classified as a target. However, multiple exemplars for the same noise region are counted only once. In effect, we are counting the image regions (as opposed to exemplars) which are classified as target regions by at least one exemplar. If a region is, in fact, a target region and some exemplar of it is called a target, that is a success. If no exemplar is so called, then a false dismissal has occurred. Finally, if the so-called target region does not, in fact, contain a target then a false alarm has occurred.

The overall classifier results consist of 6 false alarms and 3 false dismissals from the 162 target windows and 2 more false alarms from 10 non-target windows. No window contained more than one false alarm cue. Figure 3 displays the 6 (total) false dismissals. Masks of the false alarms along with their gray level windows are shown in Figure 4.

The question of how target identifications can be made in this environment of multiple exemplars, while secondary to the task of detection, is an interesting one. For each containment tree containing at least one exemplar classified as a target, we chose the target type of the exemplar with the best edge-match (E&P) score in the tree and used that target type to designate the region. In the event that the "best" exemplar was not described as a target, we labelled the object region "unknown target." Only large targets were considered, since small targets while detectable were not considered identifiable.

In a test which classified all best exemplars of large targets (55 tanks, 21 trucks, 36 APCs) the between-types confusion matrix was:

		Classified as T	Tr	A	UT
A priori	T	40	5	6	4
	Tr	6	8	7	0
	A	9	5	20	2

where "UT" is the "unknown-target" type. The 8 false alarms were classified as 1 truck, 2 APCs, and 5 small targets. Between-class confusion is high, with tanks being the most successful class. Trucks and APCs were often confused with tanks. A number of reasons can be advanced for this performance. First, tanks were the most numerous target and therefore could be identified most confidently. Second, large APCs appeared with the wooden wave deflection board in view, producing a characteristic "c" shape. No attempt was made to utilize this special knowledge. Third, the large targets appeared in only a single aspect and no generalized shape descriptors separating the different types could be extracted reliably. It seems most sensible to model the target types as three-dimensional objects, and to derive discriminators from their inherent shape and size differences from all aspects.

5. Summary

We may summarize the principal classification results as follows: The false dismissal rate of the system is less than 4%, giving a system detection rate of

96%. The false alarm rate, based on the number of false alarm regions per unit area, is 8 false alarms in 174 (128x128) windows. Assuming there are 500x800 pixels per frame and that a target occupies about 1/10 of a window, we conclude that the total processed area corresponds to about 6 frames. Thus the false alarm rate is 8/6 or 1.3 per frame. A separate test of the false alarm rate was made using a set of four 512x512 pixel frames. All available targets were detected. In addition, 4 large false alarms and 8 small false alarms were detected. However, 5 of the 8 small false alarms corresponded to fiducial marks. Moreover, one large "false alarm" (in F1) appears to be a target. In any case, 7 false alarms in 4 frames agrees well with the previous estimate of the false alarm rate.

REFERENCES

1. D. L. Milgram, Region extraction using convergent evidence, Proc. Image Understanding Workshop, April 1977, 58-64.
2. D. L. Milgram, Progress report on segmentation using convergent evidence, ibid., Oct. 1977, 104-108.
3. G. C. Stockman, Maryland Interactive Pattern Analysis and Classification System, Univ. of Maryland Computer Science Center, Technical Report 408, Sept., 1975.

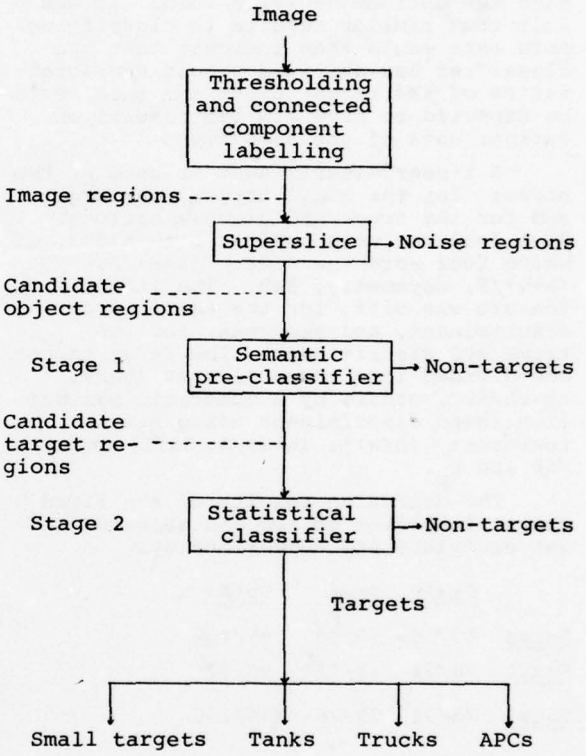


Figure 1. The classification process.

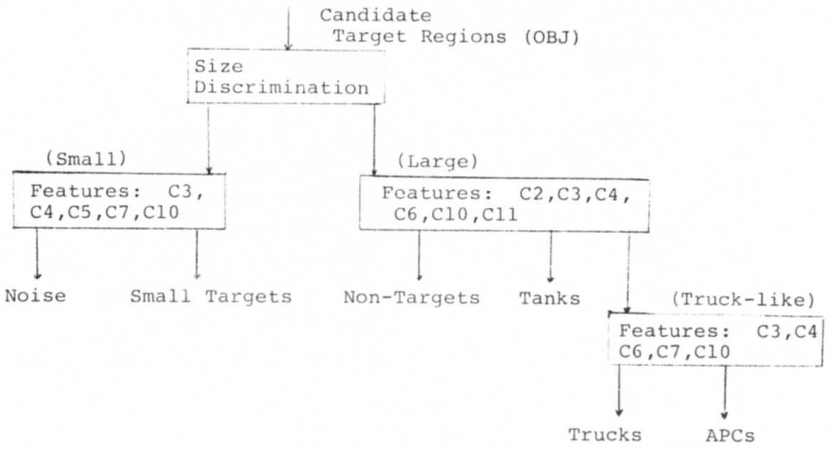


Figure 2. Stage 2 - the statistical classifier (for feature list, see Table 1).

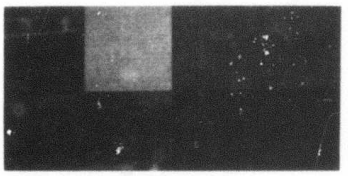


Figure 3. Six windows containing false dismissals.

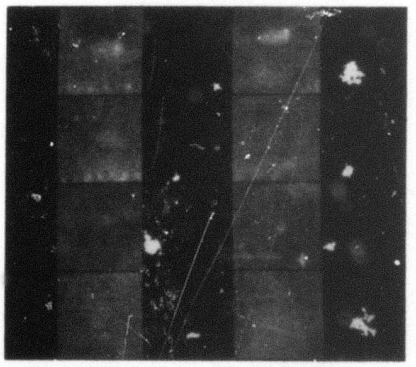


Figure 4. Eight false alarm windows with masks indicating the locations of the false alarm regions.

a. Accumulatable features per connected component

	Symbol	Meaning
1.	N	Area
2-3.	SX,SY	$\Sigma X, \Sigma Y$ - first moments
4-6.	SX^2, SY^2, SXY	$\Sigma X^2, \Sigma Y^2$, XY - second moments
7.	P	Perimeter point count
8.	E	High edge point count
9.	SPE	Total edge value on the perimeter
10.	SIG	Total interior gray value
11.	SPG	Total border gray value
12-13.	SG, SG^2	Total gray level, total squared gray level

b. Intermediate quantities

1.	X_{AVE}	$4 * \sqrt{SX^2}$
2.	Y_{AVE}	$4 * \sqrt{SY^2}$
3.	R^2	$SX^2 + SY^2$
4.	V	$S^2/N - (SG)^2 N^2$

c. Recognition features

h/w	Y_{AVE}/X_{AVE}	shape
(h/w)'	$\lvert X_{AVE} - .8 * Y_{AVE} \rvert / \sqrt{X_{AVE} * Y_{AVE}}$	shape
(h*w)/A	$X_{AVE} * Y_{AVE}/N$	shape
(h+w)/P	$(X_{AVE} + Y_{AVE} - 4)/P$	shape
diff	$(SX^2 - SY^2)/R^2$	shape
skewness	$\lvert SXY \rvert / R^2$	shape
asymmetry	$((SXY)^2 - SX^2 SY^2)/R^4$	shape
SDEV	$\sqrt{V}$	brightness
Gray level difference	SIG(N-P) - SPG/P	brightness
E&P	(Number of perimeter points at high edge local maxima)/P	brightness
E_p	SPE/P	brightness

Table 1. Features

IMAGE SEGMENTATION FOR SYNTACTIC CLASSIFICATION OF LARGE IMAGES

Raj. K. Aggarwal

HONEYWELL INC.
Systems & Research Center
2600 Ridgway Parkway
Minneapolis, Minnesota 55413

ABSTRACT

A segmentation scheme for component extraction for syntactic classification of "large" images of tactical targets is described here. It involves using prototype similarity technique iteratively, first, for target/background segmentation on the full frame at a low resolution and then for component extraction at a high resolution. Experimental results on FLIR images of tactical targets are included.

INTRODUCTION

In picture recognition problems, the number of features required is often very large, which makes the idea of describing complex patterns in terms of a (hierarchical) composition of simpler subpatterns very attractive [1]. Also, if the number of possible descriptions is very large, it is impractical to regard each description as defining a class. Consequently, the requirement of recognition can only be satisfied by a description of each class rather than by its classification. For example, the image of a tank is shown in Figure 1.

Suppose it is possible to recognize the component parts of this tank such as motor, hot vents, barrel, etc., using statistical properties of each component and their spatial relationship. The hierarchical (tree-like) structural information in this tank can be represented by a tree as shown in Figure 2.

The basic assumption in this approach is that it is easier to recognize the components instead of the tank itself. Grammatical rules can then be used to describe these trees. The grammatical rules for this example are:

TANK → RECTANGLE, HOTSPOTS, BARREL
RECTANGLE → TREAD, MOTOR, VENTS

Since different components may be seen at different target aspect angles, one could infer a general set of rules by training the classifier by tree-structures of targets viewed from different aspect angles. The general block diagram of syntactic approach to tactical target recognition problem is shown in Figure 3.

The assumption in the syntactic approach to tactical target recognition is that the images of tactical targets are "large" enough to show structure.

In the following sections, a brief description of the prototype similarity transformation and its adaptation for component extraction is given. Experimental results on FLIR images of tactical targets are also included.

SCENE ANALYSIS USING PROTOTYPE SIMILARITY

The image segmentation scheme using prototype similarity transformation can be divided into the following major steps [2]:

- Attributes
- Prototype Generation
- Threshold Selection
- Prototype Inference
- Cell Inference
- Similarity Relation

Attributes

The first step in carrying out image segmentation by the prototype similarity transformation is to decide which attributes characterize a cell. A cell can be a pixel or a certain collection of pixels depending upon the required resolution in the segmented scene. Some of the commonly used attributes are average intensity, edge feature, texture, etc. Suppose $X_1,\ldots,X_N$ are the N attributes characterizing each cell. These N atrributes may be N independent measurements on each cell or may be N functions of M ($M \geq N$) independent measurements.

Prototype Generation

For each of these N attributes characterizing a cell, a two-dimensional distribution function f (J,I) is calculated as follows: Suppose the attribute value of a cell is I. Count the number of cells in some experimentally chosen neighborhood (depending upon the resolution, size of the target, etc.,) that have attribute value J. Accumulate this sum for all the cells in the picture that have atrribute value I. This sum gives

f(J,I). Do this for all values of I and J.

The next step is to determine initial background and target prototypes using some a priori information about the scene. This can be done by locating typical background and target cells or by using some attribute information about the background/target. For example, a running motor may be the brightest part of tactical FLIR images. This information can be used to locate a target cue.

Let the target cell attribute value be A_T and background cell attribute value be A_B. Based on these two values A_T and A_B, two intervals $[A_T \cdot T_A, A_T/T_A]$ and $[A_B \cdot T_A, A_B/T_A]$ are calculated, where T_A is an empirically chosen threshold on the value of attribute A. The choice of this threshold will be discussed later. These two intervals are assumed to be disjoint. The case of overlapping intervals implies either a bad choice of target/back-round cues or a low value of threshold T_A.

These two disjoint intervals define the first two prototypes P_0 and P_1. For generating additional prototypes, consider the two-dimensional distribution function shown in Figure 4. All the cells that belong to prototypes P_0 or P_1 are zeroed (shown by hatched areas). Suppose the modified distribution function is f'(J,I). Then for each attribute value I, we have an attribute profile of neighbors. By considering each value I in the intervals $[A_T T_A, A_T/T_A]$ and $[A_B T_A, A_B/T_A]$, the accumulative attribute profiles f_{P_o} and f_{P_1} are calculated as follows:

$$f_{P_o} = \sum_{I \in [A_T T_A,\ A_T/T_A]} f'(J,I) \qquad (1)$$

$$f_{P_1} = \sum_{I \in [A_B T_A,\ A_B/T_A]} f'(J,I) \qquad (2)$$

An example of these profiles is shown in Figure 5. A maximum is located in each of these profiles. The maximum of these maxima gives the location of the next prototype interval. This corresponds to maximizing the probability of finding a neighbor that has attribute value outside the attribute intervals of previous prototypes. Suppose the attribute value is A_2. This gives rise to an interval $[A_2 \cdot A_T, A_2/A_T]$ for the prototype P_2. At this stage, there are three prototypes P_0, P_1 and P_2.

Now there are three accumulative profiles for the three intervals. The whole sequence of operations is repeated until no more prototypes can be generated. One point to remember is that the subsequent prototype intervals may overlap.

Threshold Selection

A numerical value between 0 and 1 needs to be chosen for each atrribute for defining prototype intervals. Too small a threshold leads to larger intervals and consequently fewer number of prototypes, whereas too large a threshold will lead to smaller intervals and larger number of prototypes. In the extreme cases, one one hand, we may have only two prototypes which will give rise to too many edge elements as we will see later; and on the other hand, we may have many prototypes so that each cell is similar to only one prototype giving rise to too many different objects in the scene.

For FLIR images, a typical value for the number of prototypes for each attribute is somewhere between 10 - 15. So, the thresholds can be adjusted to give the right number of prototypes.

Prototype Inference

Let $P_0,\ldots,P_N$ be the set of prototypes generated using one attribute. Each one of these prototypes has an interval on the attribute axis associated with it. Each cell in the picture is labeled by a string of prototypes it is similar to. A cell can be similar to more than one prototype as the prototype intervals can overlap. During the labeling process, a co-occurrence matrix is constructed. Each element A_{IJ} in the co-occurrence matrix, I = 0,..., N; J = 0,.,N, corresponds to the frequency that the prototypes P_I and P_J occur together in labels.

The fact that the prototype P_0 was generated by a target cue and P_1 was generated by a background cue is used to infer meaning for other prototypes. The co-occurrence matrix is used to guide the inference. Suppose A_{0I} is maximum for $I = I_1$ and A_{1J} is maximum for $J = J_1$. Depending upon which one of A_{0I_1} and A_{1J_1} is greater, either prototype P_{I1} or P_{J1} is considered for inferring its meaning. The following rules are used to infer meaning for a prototype:

- A prototype whose interval overlaps a target interval and does not overlap a background interval is a target prototype.
- A prototype whose interval overlaps a background interval and does not overlap a target interval is a background prototype.
- A prototype whose interval overlaps both target and background intervals is an edge prototype.
- A prototype whose interval does not overlap target or background interval is assigned the "meaning unknown".

Cell Inference

Each prototype in a cell label is replaced by its inferred meaning. The following string grammar is used to reduce string to a character:

$$TT \rightarrow T$$
$$EE \rightarrow E$$

$$BB \rightarrow B$$
$$TB \rightarrow E^*$$
$$TE \rightarrow T$$
$$BE \rightarrow B$$
$$E^* \alpha \rightarrow E^* \quad , \alpha \in \{T, B, E, E^*\}$$

where

T =>target cell
B =>background cell
E*=>strong edge cell
E =>weak edge cell

Similarity Relation

Based on each attribute, using the above described procedure, a meaning can be assigned to each cell of the picture. Thus, each cell has a string of cell meanings, the length of the string being equal to the number of attributes. In order to assign a unique meaning to the cell, a relationship between the various attributes is needed. Here, this is called a similarity relation. One simple example of a similarity realtion is that even if a cell is different in one attribute, it is different. This would mean that a cell should be assigned the same meaning by all the attributes before it is assigned that meaning. Otherwise, the cell is classified as "meaning unknown". A more complex relationship can be devised depending upon the type of imagery, type of attributes, etc.

COMPONENT EXTRACTION

The general block diagram for component extraction of tactical targets through the iterative application of prototype similarity is shown in Figure 6. First target/background segmentation is performed on a full frame at low resolution using prototype similarity transformation. Any a priori information about the scene is passed on to the segmentation scheme in the form of cues. Chain coding [3] is then used to isolate the target region of interest. The prototype similarity transformation is used on this region at an increased resolution for component extraction.

EXPERIMENTAL RESULTS

The prototype similarity transformation was tried on FLIR images of tactical targets. The technique was first tried on full frames (520 pels x 480 pels) and on the isolated targets to extract components. The target center and its approximate size were recorded during digitization. The 8-bit digitized data was scaled down to 100 grey levels to cut the computer memory requirements for storing joint distribution function.

A cell was defined as 2 pels x 2 pels for component extraction and as 4 pels x 4 pels for target/background segmentation. A neighborhood of 3 cells x 3 cells was used in both cases for calculating the joint distribution function. The only attribute used was the average intensity over the cell. A threshold of 0.85 was used to define the prototype intervals. Approximately the same number of prototype (~ 10) were obtained for both cases.

The results are shown in Figures 7 - 10. In each set of three photographs, the top picture shows the original, the middle one the target/background segmentation on full frames and the bottom one, the extracted components.

ACKNOWLEDGMENTS

The author wishes to thank Mr. Rod Larson and Dr. M. Geokezas of the SIP group at Honeywell for the helpful discussions and encouragement. He also wishes to thank Mr. R. Touchberry for software implementation and Ms. Tremel for skillful typing in preparing the manuscript.

REFERENCES

1. Fu, K. S.: Syntactic Methods in Pattern Recognition. Academic Press, 1974.

2. Aggarwal, R. K.: Image Segmentation Using Prototype Similarity. SPIE Proceedings, March 1978.

3. Freeman, H.: Computer Processing of Line Drawing Images. Computing Surveys 6, pp 57 - 97, 1974.

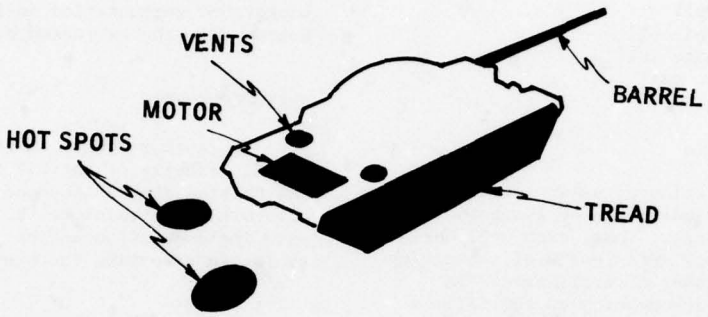


Figure 1. Image of a tank.

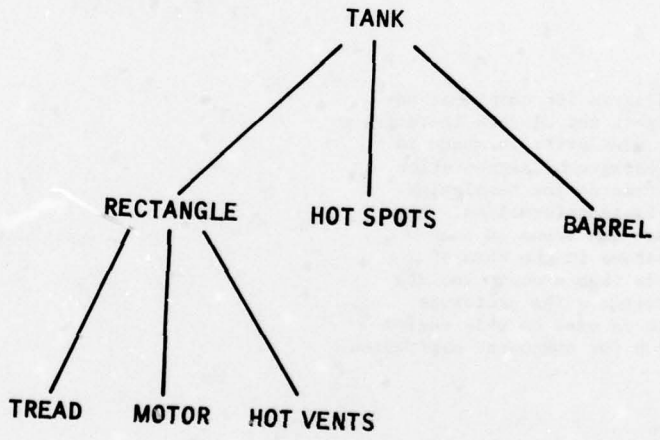


Figure 2. Hierarchical structural description of the tank shown in figure 1.

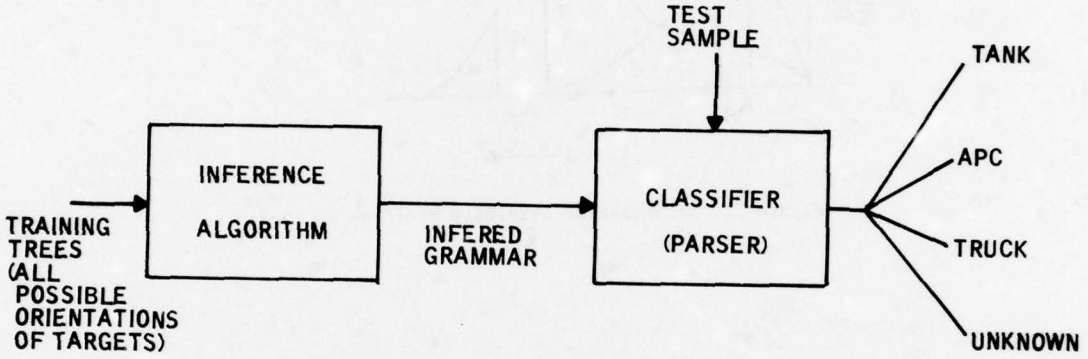


Figure 3. Syntactic approach for tactical target recognition.

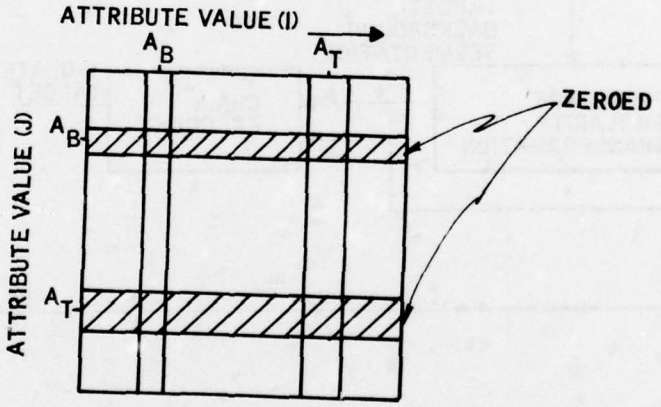


Figure 4. Two-dimensional distribution function f'(J, I).

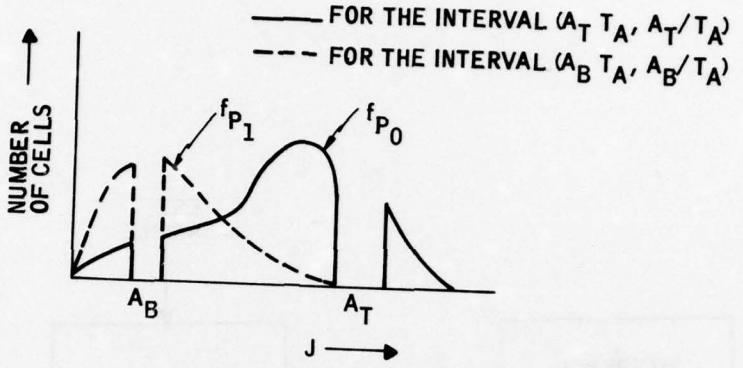


Figure 5. Accumulative attribute profiles f_{P_o} & f_{P_1}.

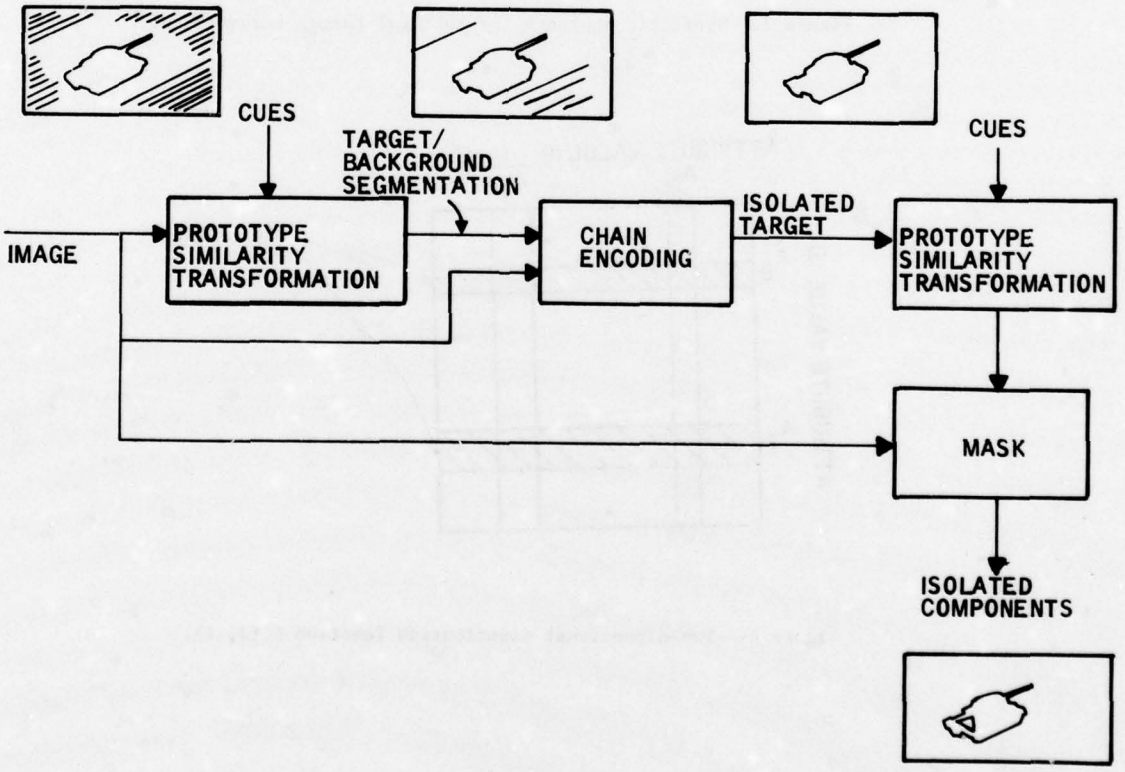
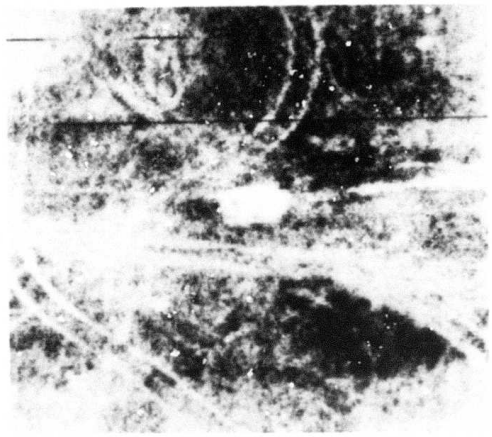
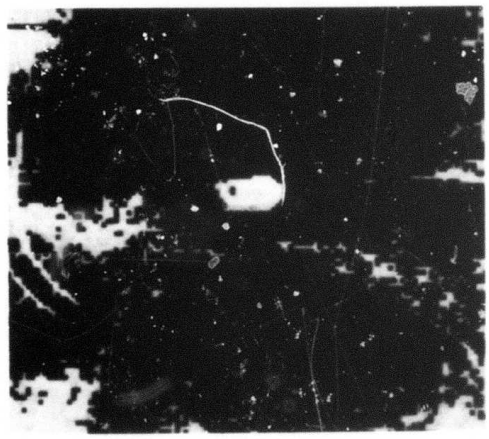


Figure 6. Component extraction of tactical targets through iterative use of prototype similarity.



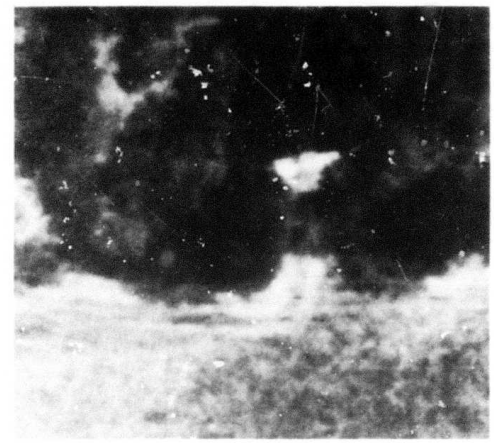
(A)



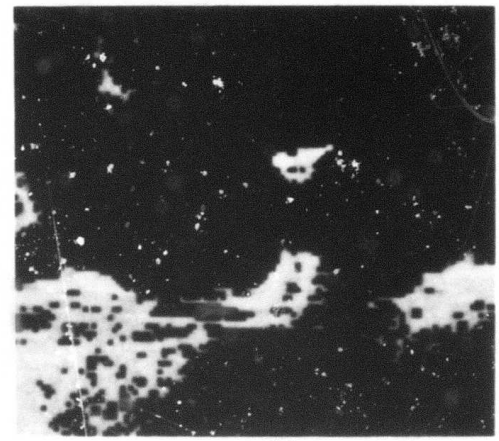
(B)

(C)

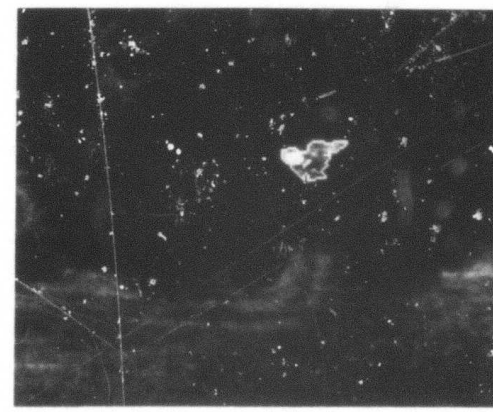
Figure 7. (A) A tank in the open.
(B) T/B segmentation on full frame.
(C) Extracted components.



(A)

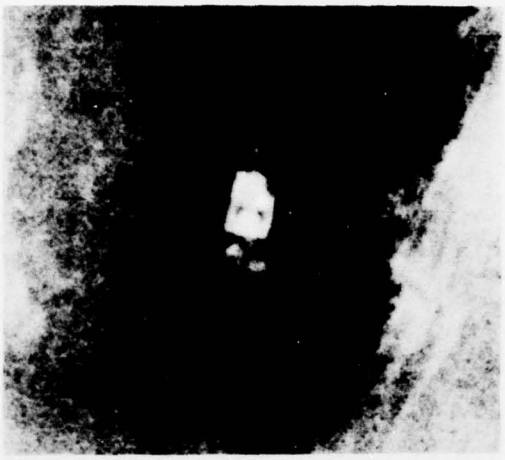


(B)



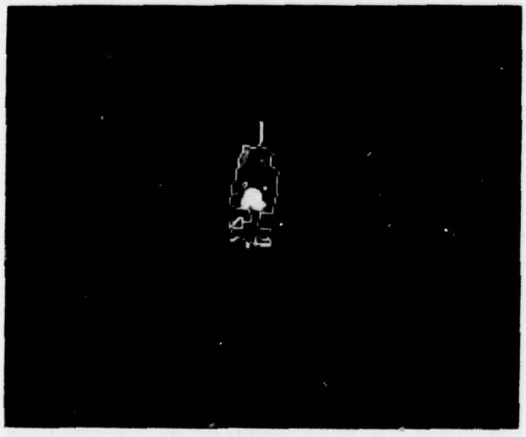
(C)

Figure 8. (A) A tank in the woods.
(B) T/B segmentation on full frame.
(C) Extracted components.



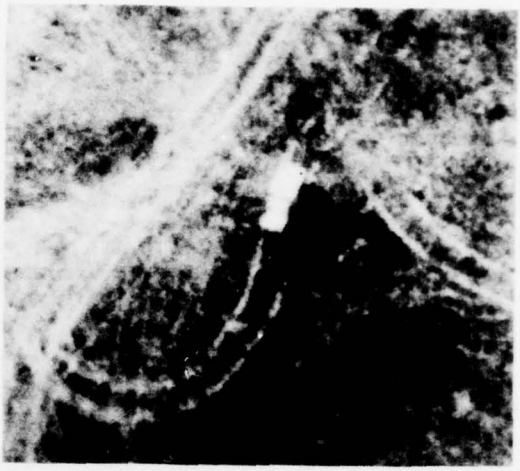
(A)

(B)

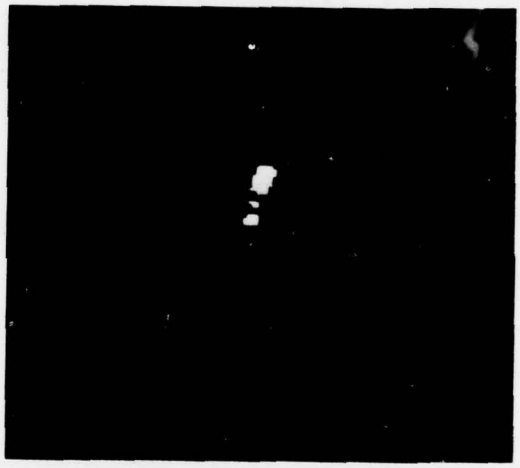


(C)

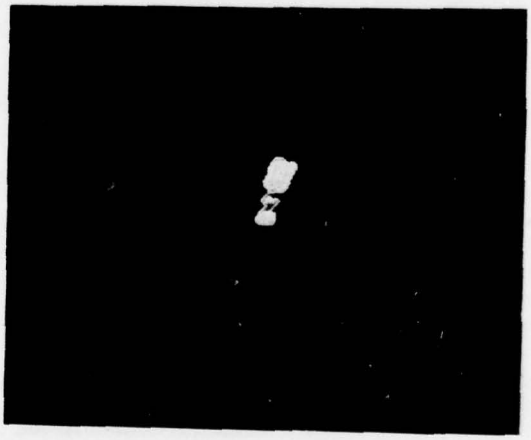
Figure 9. (A) A tank in the open.
(B) T/B segmentation on full frame.
(C) Extracted components.



(A)



(B)



(C)

Figure 10. (A) A truck in the open.
(B) T/B segmentation in the full frame.
(C) Extracted components.

ADAPTIVE THRESHOLD FOR AN IMAGE RECOGNITION SYSTEM - BACKGROUND RESULTS AND CONCLUSIONS

D. Serreyn and R. Larson

Honeywell Inc.
Systems & Research Center
2600 Ridgway Parkway
Minneapolis, Minnesota 55413

Abstract

This paper augments the paper entitled "Adaptive Threshold for an Image Recognition System"* given at the DARPA workshop October 1977. The purpose is to provide additional information about the simulation studies that were performed as well as the results obtained with the Autoscreener (ATSS) with autothreshold. This paper consists of the background and concept of the autothreshold, specific examples and relationships used in the simulation studies, the hardware results, and the performance evaluation.

Autothreshold Concept

Prior to incorporating an automatic thresholding feature, the ATSS had eighteen dials and switches that were required to be set by the operator. Some were set once but others depend upon the input image and had to be adjusted periodically, if not continuously. This was done by the operator using the first level feature display as a feedback mechanism to optimize the threshold levels. The basic purpose of autothreshold is to eliminate the need for all the manual adjustment such that the operator can perform other duties encountered on a tactical mission.

The basic concept behind autothreshold is that it makes the autoscreener adaptive to changing scene intensity and contrast levels. It does this automatically be determining the threshold for edge and high/low intensities on a scan line basis.

The overall concept for autothreshold is shown in Figure 1. Each box will be discussed in greater detail later; but briefly, the function of each box is as follows: Raw video is passed through a low pass smoothing filter. This limits the bandwidth of the noise. The smooth data is an input to both the edge filter and the bright filter. The edge filter generates the magnitude of an edge in an image from which an edge threshold is determined. The output EDGE is a logical signal used the the Autoscreener and is obtained by comparing the analog edge signal with its threshold. The bright filter determines the background level and is a basis for determining the bright threshold. The BRIGHT signal is also a logical signal which is used by the Autoscreener for further processing of man-made objects.

*"Adaptive Threshold for an Image Recognition System", D. Serreyn and R. Larson, DARPA Image Understanding Workshop Proceedings, October 20-21, 1977, pp. 73-73.

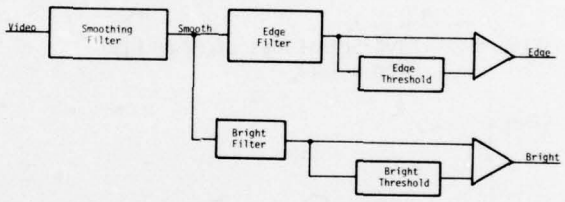


Figure 1. Functional Block Diagram - Autothreshold

Simulation Studies

The autothreshold was simulated as shown in Figure 1. The low pass smoothing filter limits the bandwidth of the noise that enters into the edge and bright filters. The smoothing filter was a weighted average based upon the following equation:

$$A_{ij} = \frac{1}{16}\Big[I(i-1,j-1) + 2\,I(i-1,j) + I(i-1,j+1)] + 2I(i,j-1) + 4I(i,j) + 2I(I,j+1) + I(i+1,j-1) + 2I(i+1,j) + I(i+1,j+1)\Big]$$

where I is the video intensity. This filtered video is then the input to the edge filter and to the bright filter. Figure 2 is a sample of five scan lines of FLIR video over a tank. Figure 3 is the resultant smooth data.

The two dimensional SOBEL edge filter was simulated as follows:

$$H_{j-1} = I(i-1,j-1) + 2I(i,j-1) + I(i+1,j-1)$$
$$H_{j+1} = I(i-1,j+1) + 2I(i,j+1) + I(i+1,j+1)$$
$$V_{i-1} = I(i-1,j-1) + 2I(i-1,j) + I(i-1,j+1)$$
$$V_{i+1} = I(i+1,j-1) + 2I(i+1,j) + I(i+1,j+1)$$

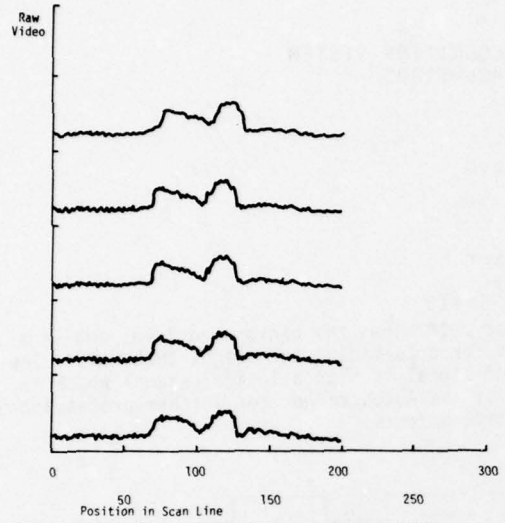


Figure 2. Five Scan Lines of Raw Video Over a Tank

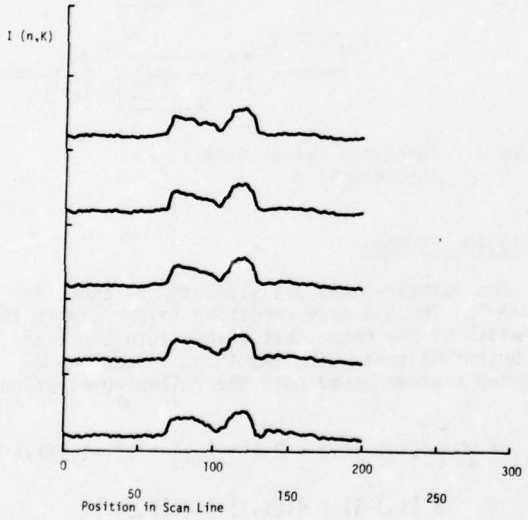


Figure 3. Smooth Video of Data in Figure 2

where H and V are the horizontal and vertical components. Then, the edge value associated with $(i,j)^{th}$ pixel is

$$E(i,j) = |H_j| + |V_j|$$

$$= |H_{j+1} - H_{j-1}| + |V_{i+1} - V_{i-1}|$$

Figure 4 is the Sobel edge for the data shown in Figure 3. Superimposed on the edge data is the adaptive threshold. The edge threshold is

$$E1 = K * \overline{E_{n-1}}$$

where E1 is the threshold, $\overline{E_{n-1}}$ is the previous scan line edge average and K is an optimum constant statistically determined. Figure 5 is the EDGE output for a tank scene.

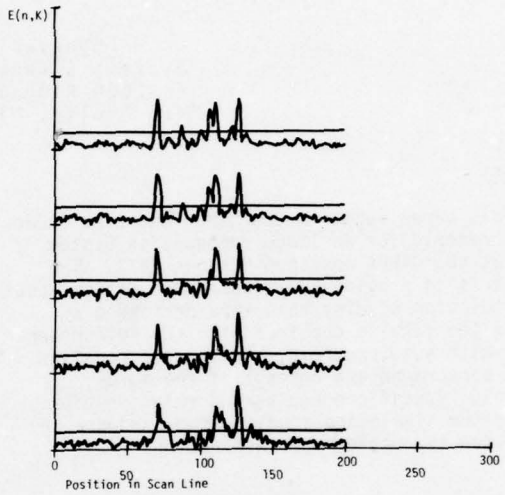


Figure 4. Edge with Threshold of Data in Figure 3

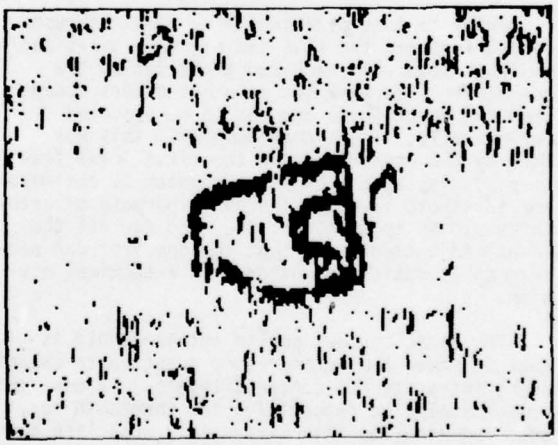


Figure 5. EDGE Output for the Tank Scene

As was mentioned previously, the smoothed video is fed into the bright filter. A primary function of the bright filter is to estimate the background in each frame. This background must be continually updated as the image is scanned by a recursive filter and filter updating logic.

The first decision we make is to determine if there is large contrast between scan lines on a pixel by pixel basis. This is compared to a threshold TLIM. TLIM is an average absolute difference between the present and previous scan line multiplied by a constant, that is:

$$TLIM = \frac{\alpha\Sigma|I_{i+1,\,j} - I_{i,j}|}{Num}$$

α = constant
Num = number of pixels in one scan line

The background estimate J is built up over several scan lines. The background is updated throughout most of the scan line and is defined as

$$J_{i,j} = \beta J_{i-1,j} + (1-\beta)\, I_{i,j}$$

We do not update the background estimate over areas of large contrast. When not updating, $J_{i,j} = J_{i-1,j}$. The value of $I_{i,j} - J_{i-1,j}$ is compared to a value SLIM where SLIM is defined by

$$SLIM = \frac{\alpha\Sigma|I_{i,j} - \hat{J}_{i,j}|}{Num}$$

where $\hat{J}_{i,j}$ is $J_{i,j}$ filtered by a Hamming window.

In summary, this pixel classification scheme is as follows:

if $|I_{i,j} - I_{i-1,j}| > TLIM$

or

if $|I_{i,j} - J_{i,j}| > SLIM$

don't update the background filter.

Also, when updating

$$J_{i,j} = \beta J_{i-1,j} + (1-\beta)\, I_{i,j}$$

otherwise, $J_{i,j} = J_{i-1,j}$

The pixel classification for the five scans shown in Figure 3 is shown in Figure 6.

Once the background $J_{i,j}$ is determined, it is subtracted from $I_{i,j}$ to give a zero reference. Hence, $Z_{i,j} = I_{i,j} - J_{i,j}$ is data with zero reference that must be thresholded. $Z_{i,j}$ is compared to a variability threshold EPSI. EPSI is defined as

$$EPSI = \mu * VARJ = \frac{\mu}{(1-\beta)}\ \Sigma|J_{i,j} - J_{i-1}|$$

μ: a constant determined during simulation

Figure 6. Pixel classification (High is Don't Update)

The background estimate for the smoothed video (Figure 3) is shown in Figure 7. The resultant video, to be thresholded, is shown in Figure 8. Figure 9 is the BRIGHT output for the sample tank image. The EDGE and BRIGHT data are logically combined to produce features used by the autoscreener for the detection of man-made objects.

Hardware

The edge and bright filters were implemented as part of the control unit. The smoothing filter is accomplished by using the scan converter in an integrating or averaging mode. The edge filter block diagram implementation is shown in Figure 10. Each scan line delay consists of 2 Fairchild CCD321 (455-910 element) delay lines. The pixel delays are obtained from selected taps of a Reticon TAD-32 (a 32 tapped analog delay line). The low pass filter smooths the edge output.

The threshold value E1 is determined by integrating the edge over the previous scan line.

$$E1 = K * \overline{E_{n-1}}$$

where E1 is the threshold, K is a constant and E_{n-1} is the previous scan line edge data. E_n is compared to a threshold. When E_n exceeds E_1, the logical EDGE signal is created which is used by the rest of the autoscreener.

A block diagram of the background estimate and bright threshold implemented is shown in Figure 11. The background estimate is a recursive filter whose time constant depends upon the parameter β.

The low pass filter limits the clock noise coming out of the CCD line delay.

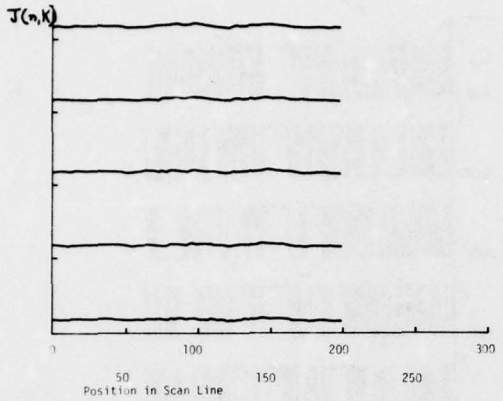


Figure 7. Background Estimate of Data in Figure 3

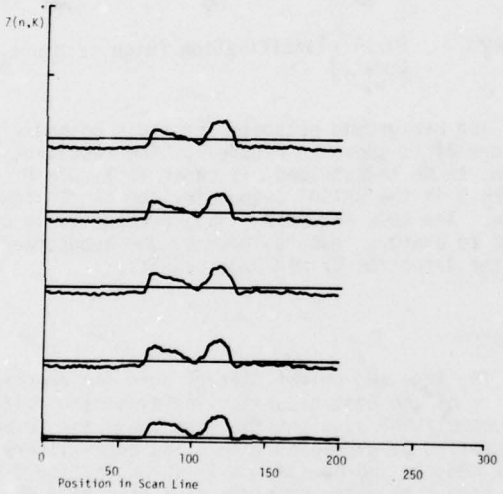


Figure 8. "Video-Background" of Data in Figure 3

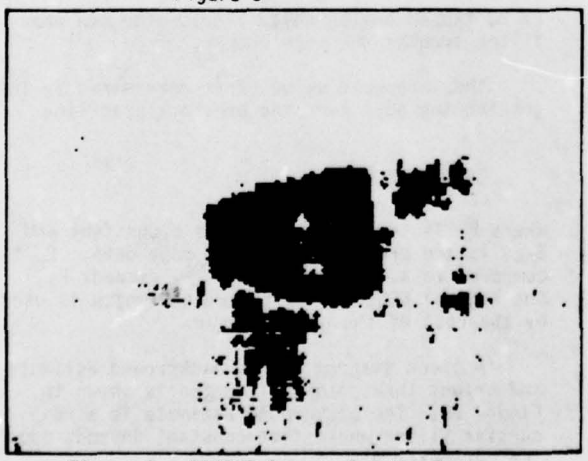
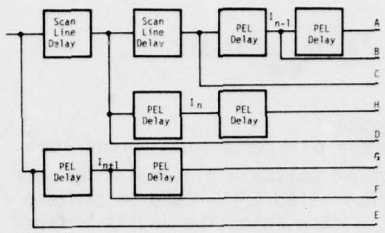


Figure 9. BRIGHT Output for the Tank Scene



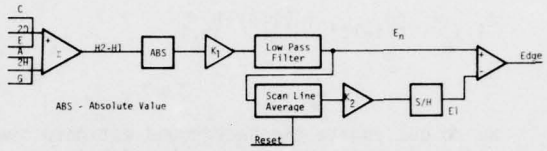


Figure 10. Autothreshold Edge Hardware Implementation

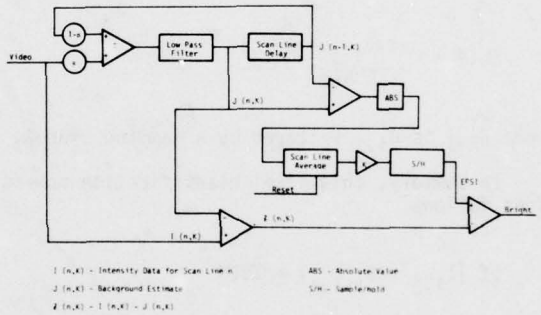


Figure 11. Autothreshold Bright Hardware Implementation

The threshold EPSI is based upon the absolute difference between background estimate, that is

$$EPSI = K * \overline{|J_{n,k} - J_{n-1,k}|}$$

where K is a constant and $J_{n,k}$ and $J_{n-1,k}$ are background estimates. The threshold is based upon the previous scan line.

In addition to the clock noise, the chosen CCD's generated periodic noise due to dark current. The period of the noise was approximately equal to one-eight (1/8) of the 455 elements. Two of the 455-910 line delays were used to make up the 1820 pixel line delay and the noise was in all the devices. We discovered that the delay line serpentine implementation was the cause of the noise. The corners exhibited excessive dark current noise that is especially noticeable when the clock is stopped for a period of time. A new device is being designed by the vendor which is anticipated to eliminate this problem.

A representative sample of the results from the hardware is presented in the following figures. The pictures were taken off the first level displays which are a part of the Autoscreener. Figure 12 is the raw video with a symbol directly below the target. Figure 13 is the horizontal edge component. Figure 14 is the pixel classifier output even though it is not in the feedback loop of the hardware. Figure 15 is the bright or high intensity image. Figure 16 is the F1 signal which is the result of logically combining the EDGE and BRIGHT signal.

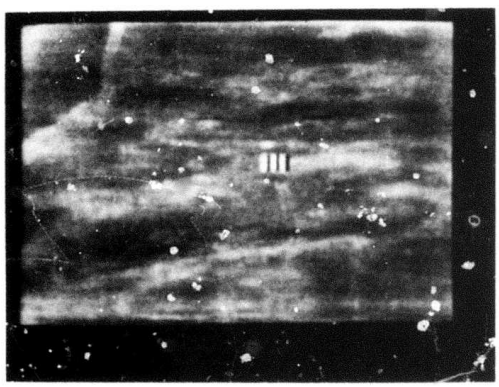


Figure 12. Raw Video Scene

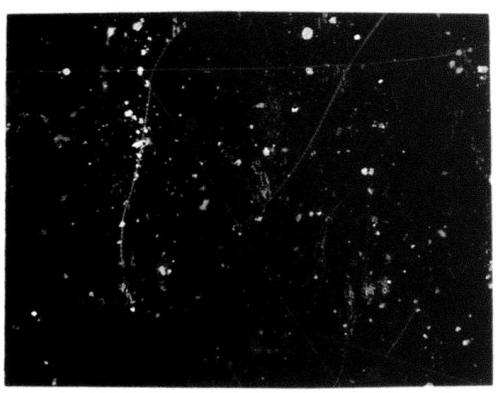


Figure 13. Edge Output for Raw Video Scene

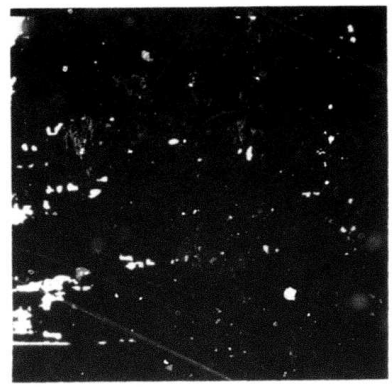


Figure 14. Pixel Classifier Output

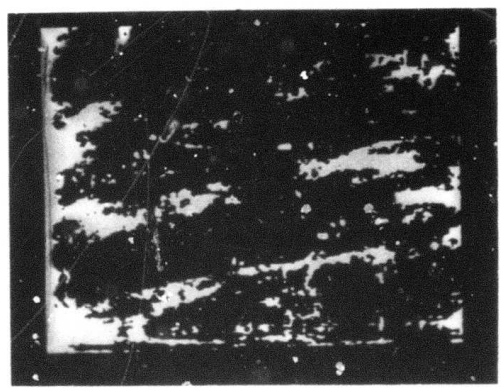


Figure 15. BRIGHT Output for Video Scene

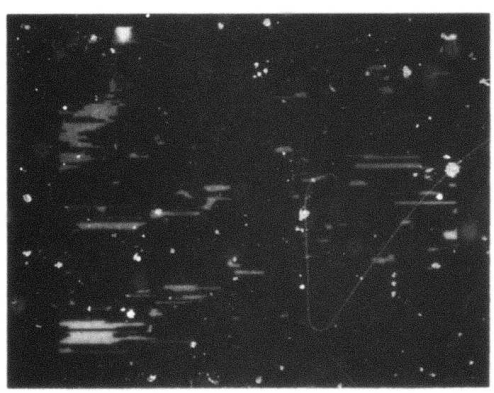


Figure 16. F1 Signal for Video Scene

Performance Evaluation

The performance of the autoscreener/ FLIR system with autothreshold to detect man-made objects was evaluated utilizing about 1150 frames of FLIR imagery.

Prior to testing the classifier was trained using 164 frames which contained 2023 candidate objects (158 man-made and 1864 nuisances).

The scoring consisted of playing back the video frames from a video disc. A symbol was generated if a sector (1/16 of a frame) contained one or more objects classified as targets, these symbols were superimposed on the image and displayed on the monitor and at the same time were recorded and then scored in order to evaluate the performance. If a sector contained one or more MMO's and they were not detected (no symbol was displayed) this represented a missed MMO's sector. On the other hand, if a sector contained no MMO's and a symbol was displayed, this was considered a sector with a false alarm.

The probability of MMO detection P_D is the ratio of the number of detected sector with MMO's to the total number of sectors with MMO. The probability of a miss is P_M - 1-P_D. The probability of false alarm P_{FA} is the ratio of the number of sectors with false alarms to the total number of sectors without MMO's. This result is shown in Table 2.

TABLE 2

Total Number of Sectors	18496
Number of Sectors with MMO's Missed	99
Detected Sectors with MMO's	1027
Sectors with False Alarms	747
Sectors with MMO's	1126
Sectors without MMO's	17370

$$P_D = \frac{1027}{1126} = 91.2\%$$

$$P_{FA} = \frac{747}{17370} = 4.3\%$$

$$P_M = \frac{99}{1126} = 8.8\%$$

This point is plotted in Figure 17. The result of 91.2% probability of detection and 4.3 probability of false alarm is very nearly the same as FLIR without auto-threshold.

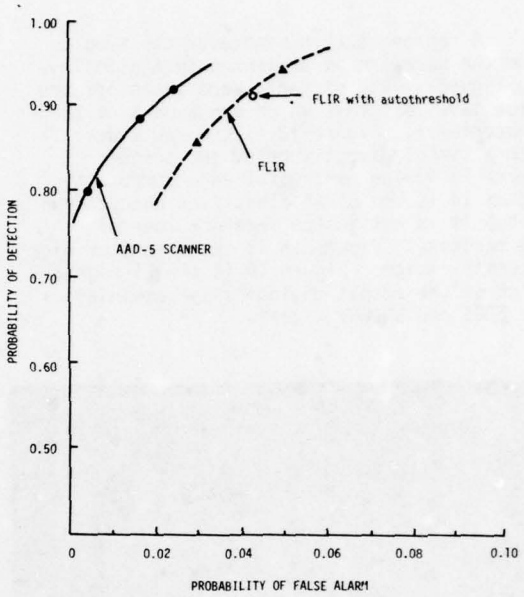


Figure 17. Autoscreener/Autothreshold Performance: Sector Basis

Conclusions

At the completion of this program, the following additional conclusions are made:

1. We have established the feasibility in in detecting man-made objects with an Autoscreener with autothreshold. We achieve a 91.2% detection probability and 4.3% false alarm probability.

2. The scan line delays needed for two dimensional processing must be made to operate in a start/stop mode of operation with little degradation in signal.

3. A more robust classifier which would screen out objects based upon additional shape and size feature will reduce the number of false alarms. This concept called secondary screening would look at only those objects classified as MMO's by the present classifier.

4. The false alarm probability and scoring should be changed to include time or rate such as the average number of false alarm/frame processed or false alarms per second.

SESSION V

PROGRAM REVIEWS

BY

PRINCIPAL INVESTIGATORS

MIT PROGRESS IN UNDERSTANDING IMAGES

Patrick H. Winston

The Artificial Intelligence Laboratory
Massachusetts Institute of Technology

In the previous proceedings, we stressed the key issue of representation. In particular, we described research of Horn and his collaborators using the reflectance map *as a tool for working with satellite images, and we described work of Marr and his collaborators using the* primal sketch, *the* 2 1/2 D sketch, *and* generalized cones *to work toward a comprehensive theory of recognition.*

Here, we cite some of the problems having to do with using real satellite images, we report on the development of a machine for rapid primal-sketch computation, and we take note of some new work on depth vision and representation.

Registering Images And Making Albedo Maps

Horn has demonstrated a method for registering aerial photographs with terrain models that potentially yields registration accuracy in the subpixel range. The method works by comparision of the given photograph with a synthetic image produced using the corresponding terrain model. Given registration, it then becomes possible to do several things starting with the same reflectance-map based technology. For example, we have made some images in which the hue reflects the ratio of real image intensity to synthetic image intensity, and we believe these images provide a good index to ground cover. This ratio does not depend much on sun position, unlike other measures used up to now. We call an image made up of these ratios an albedo map.

To make really useful albedo maps, however, we have found it necessary to solve several subproblems. Dealing with real satellite images requires the solution of several problems of the sort that escape notice when thinking is done in terms of idealized domains. One of these is the problem of introducing cast shadows into the synthetic image. This has been done.

Destriping, Transforming Coordinates And Finding Where The Sun Is

Other problems inherent in the use of real satellite images include those introduced by the characteristic flaws of satellite images, by the need for care in dealing with coordinate transformations, and by the need to know accurately where the sun is.

First, corrections to satellite images must be made to account for differences in the transfer functions of the several sensors used. The paper of Horn and Woodham, elsewhere in the proceedings, gives the results of their work on the problem. The paper describes a method that uses statistics obtained from the sensors themselves, together with an assumption that the probability distribution of the scene radiance seen by each image sensor is the same. Using the method, they have sucessfully removed the striping effects seen commonly in satellite photographs.

Next, coordinate transformation is necessary in order to do proper registration of satellite photographs against earth surface models. (The surface models considered are in the form of surface elevations on a grid of points.) Consequently, Horn and Woodham have developed an affine transformation between the coordinates of Multispectral Scanner images produced by the LANDSAT satellites and the coordinates of a system lying in a plane tanget to the earth's surface near the subsatellite point.

Finally, as Horn has stressed in his papers, the appearance of a surface depends dramatically on how it is illuminated. In order to interpret satellite and aerial imagery properly, it is therefore necessary to know the position of the sun in the sky. Horn has developed relatively straightforward methods for doing so with more than enough accuracy for image understanding purposes.

Primal Sketch Hardware

Much of Marr's image understanding work requires the computation of a so-called primal sketch. The primal sketch is a rich symbolic description of the important features exhibited by an image, edges and blobs in particular. Creating such a symbolic description requires a great deal of convolution. Consequently, there has been a need for fast image convolution hardware. We have just completed and have begun to test ICON, a first prototype of such a convolver.

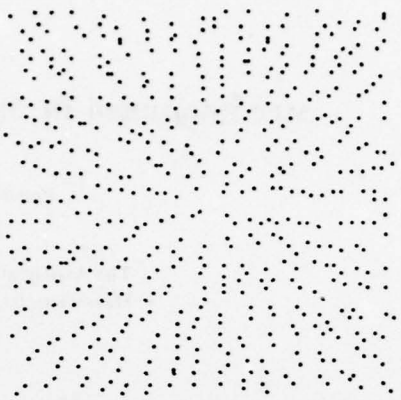
ICON combines a pipelined VLSI multiplier with a fast bipolar image cache. Approximately 120 Schottky MSI and LSI IC's are used. The device is connected as a peripheral to the LISP Machine and is driven by microcode.

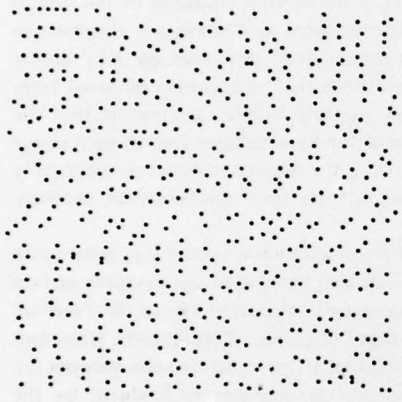
ICON can convolve a 16 x 16 mask against an image point in 50 microseconds. An entire 512 by 512 image mask convolution

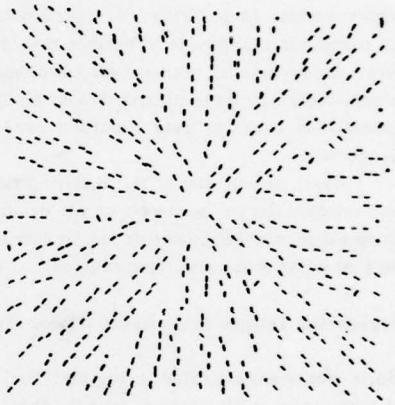
can be done in less that 15 seconds. This represents more than an order of Imagnitude speedup over PDP-10 performance.

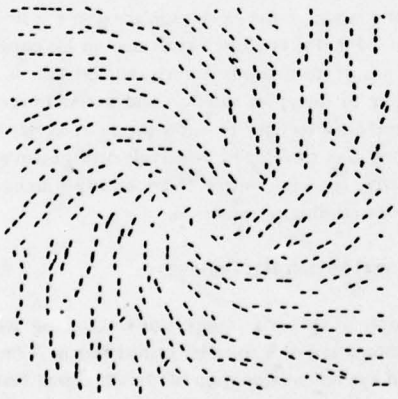
Local Image Structure

Kent Stevens has completed a study of textures that involve a sense of "flow" or rough "parallelism." Locally parallel dot patterns were used extensively in testing his ideas. Such dot patterns are transformed by primal-sketch machinery into a collection of *place tokens*. Stevens' parallelism algorithm then constructs *virtual lines* that radiate from each place token in the image to its neighbors. The orientations of these are histogrammed, and the candidate virtual line that corresponds most closely in orientation with the histogram's maximum is selected.









In the figure, we show input dots on the left and the virtual lines corresponding to derived local parallelism on the right. The algorithm handles place tokens derived from edge features as well as from dots, as is required in working with natural images.

Stereo

David Marr in conjunction with Tomaso Poggio (of the Max Planck Institute for Biological Cybernetics) completed a new study of stereo vision. The resulting algorithm consists of five steps: (1) Each image is filtered with bar masks of four sizes that vary with eccentricity; the equivalent filters are about one octave wide. (2) Zero-crossings of the filter outputs are localized, and positions that correspond to terminations are found; (3) For each mask size, matching takes place between pairs of zero-crossings or terminations of the same sign in the two images, for a range of disparities up to about the width of the mask's central region; (4) Wide masks control vergence movements, thus causing small masks to come into correspondence; (5) When a correspondence is achieved, it is written into a dynamic buffer, the 2 1/2 D sketch. In addition to being satisfying from the automatic image understanding point of view, Marr has shown that the algorithim provides a

theoretical framework for most existing psychological data about stereo.

Eric Grimson has finished a computer implementation of the algorithm that is highly succesful in computing disparity from a stereo pair of photographs taken of natural scenes. The implementation has been found to be an important research tool in revealing phenomena concerning the convolution of natural images with bar masks.

Currently, we are busy testing the algorithm, as well as turning towards issues concerning the "filling in" of depth information where it cannot be recovered directly from the image. These issues interface with more general issues concerning the representation of spatial information.

2 1/2 D Representation

Shimon Ullman, Eric Grimson, and Kent Stevens have made progress on three aspects of the problem of representing information about surfaces. Ullman has tried to tie Horn's work on judging shape from shading to the portion of the 2 1/2 D sketch that represents local surface orientation; Grimson has worried about how local depth information can best be represented in the sketch; and Stevens has addressed the problem of inferring surface orientation from an object's boundary contours. The 2 1/2 D sketch is proposed to be a representation in which these various sources of information are integrated into a single, coherent perception of visible surfaces.

References

Berthold K. P. Horn and Brett L. Bachman, "Using Synthetic Images to Register Real Images with Surface Models," AIM-437, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1977.

Berthold K. P. Horn and Robert J. Woodham, "LANDSAT MSS Coordinate Transformations," AIM-465, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1978.

Berthold K. P. Horn, "The Position of the Sun," Working Paper 162, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1978.

Berthold K. P. Horn and Robert J. Woodham, "Destriping Satellite Images," Working Paper 163, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1978.

David Marr, "Representing Visual Information," AIM-415, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1977.

David Marr and Keith Nishihara, "Representation and Recognition of the Spatial Organization of Three-dimensional Shapes," *Phil. Trans. Roy. Soc. B. 275*, in publication.

Kent Stevens, "Computation of Locally Parallel Structure," *Biological Cybernetics*, in publication. Also available as AIM-392, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1978.

ALGORITHMS AND HARDWARE TECHNOLOGY FOR IMAGE RECOGNITION

Computer Science Center
University of Maryland
College Park, MD 20742

Systems Development Division
Westinghouse Corporation
Baltimore, MD 21203

ABSTRACT

This report lists the principal accomplishments on Contract DAAG53-76-C-0138 (DARPA Order 3206), covering the period 1 May 1976 through 28 February 1978. This research was monitored by the U. S. Army Night Vision Laboratory, Ft. Belvoir, VA; project monitors were Mr. John S. Dehne and Dr. George R. Jones.

1. Design and implementation of a comprehensive algorithm for object recognition in FLIR imagery with a detection rate above 95% and a false alarm rate between 1 and 2 false alarms per frame.
2. Fabrication and testing of a CCD sorter chip capable of operating at 3 megapixels/sec. The sorter function is a crucial step in several image operations including histogramming, median filtering, non-maximum suppression and connected component coloring.
3. Investigation of the cost, performance and constraint tradeoff in implementing a target cueing algorithm in CCD (charge-coupled device) technology. The resulting design is within specifications for usage in smart sensors.
4. Development of the "Superslice" algorithm for reliable region extraction based on the cooccurrence of border points of regions with locally maximum edge detector responses. This is an important example of the use of convergent evidence to strengthen assertions.
5. Design and analysis of statistical models for threshold selection, image operation response prediction, and optimal edge detection.
6. A new method for adaptive quantization of an image which reduces the number of gray levels present using only the histogram.
7. Comparison of image smoothing methods, including median filtering.
8. A study of shrink/expand noise cleaning schemes, including a local min/max method which cleans the image prior to thresholding.
9. Evaluation of a variety of edge detectors and the development of a reliable method for edge thinning.
10. Construction of a "fuzzy" thinning algorithm which allows thinning to occur in gray level images prior to thresholding.
11. Development of methods for threshold selection based on gray level and gradient value.
12. Generalization of thresholding to the multiple object class environment with the ability to predict appropriate (gray level, gradient value) segmentation regions for the object classes present.
13. A variable thresholding scheme which produces a binary (or ternary) representation of an image.
14. An extension of threshold selection for sequences of images.
15. Simplification of the logic of the standard connected component coloring algorithm and its extension to produce a chain encoding of the component boundary in a single pass.
16. Implementation of Hyperslice: a recursive segmentation which improves the Ohlander region extraction method.
17. An algorithm for region tracking in image sequences using dynamic programming.
18. Comparison of features for target recognition.
19. Construction of a hierarchical classifier for target detection and recognition.
20. Development of Viewmaster - a software aid to assist in the construction of image processing programs.

IMAGE UNDERSTANDING RESEARCH AT CMU: A Progress Report

Raj Reddy
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213
April 1978

INTRODUCTION

The primary objective of our research effort is to develop techniques and systems which will lead to successful demonstration of image understanding concepts over a wide variety of tasks, using all the available sources of knowledge. We are focusing our attention on three areas of research. First, we are developing an integrated concept demonstration of an image understanding system. The long-term goal of this research is to understand how knowledge can be used in the image interpretation process to produce systems which are 2 to 3 orders of magnitude more cost-effective than current systems. Over the next three years we expect to investigate how knowledge of maps, size and shape of landmarks such as buildings and rivers, and contextual relationships can be used in the interpretation of satellite images of the Washington, D.C. area and color scenes of downtown Pittsburgh.

The second area of research is the development and validation of concepts for computer architectures used in image understanding. The long-term objective of this research is to develop new computer architectures which will make low-cost image processing a serious possibility. We plan to evaluate the desirability of new processor designs and new instruction sets for image processing applications.

The third area is the development of intelligent interactive aids for tasks such as photo interpretation and map generation. Many of the same techniques which are useful in automatic interpretation are applicable in this area, except that in this case the human being provides the goal direction. The availability of intelligent assistants capable of examining large image data bases and retrieving desired information is expected to significantly improve human productivity in tasks such as photo interpretation and cartography.

The following is a brief summary of our work over the last six months.

SYSTEMS AND TASKS

The image understanding research at CMU uses a DEC System 10/80, C.mmp (a 16 processor multi-mini computer system), Cm* a large asynchronous multiprocessor (50 LSI-11 processors), and a dedicated MIPS (Multi-sensor Image Processing System) computer.

Our present plans are to attempt to interpret uncontrived arbitrary images representing different views of the downtown Pittsburgh area (a 3-D world), and aerial and satellite views of the Washington, D.C. area (a 2-D world). The world models for these tasks are expected to be generated incrementally over the next few years.

KNOWLEDGE REPRESENTATION AND SEARCH

At present we are developing the following knowledge sources for the downtown Pittsburgh task: a 3-D model of the downtown Pittsburgh area, knowledge about building structures and textures, knowledge about local refinements given coarse recognition (e.g., detecting cars in roads and trees and bushes next to roads), knowledge about shadows occlusions and highlights, and so on. Given our basic approach of iterative refinement of knowledge, we will start with simple versions of these knowledge sources, and refine them as we observe their limitations when applied to different scenes.

Since the last Workshop the ARGOS Image Understanding System has become fully operational. A Ph.D. thesis on this system by Steve Rubin is expected within the next few months (Rubin, 1978). The system has an internal three-dimensional model of the city of Pittsburgh which contains over fifty buildings, rivers, bridges, and other geographic features. Using this model, ARGOS has been trained to recognize five common views of the city from the north-west, north-east, south-west, due west, and downtown at the intersection of the three rivers. Seven digitized images of the city were used for training of the spectral characteristics. Another eight were reserved for test purposes. ARGOS was able to label all eight with 60% accuracy on a pixel basis. Further, 20% of the pixels are unlabeled and approximately 20% are incorrectly labeled. These results are expected to be significantly improved with systematic error analysis.

Much of the error is attributable to inflexibility in the training data. For example, the recognition of building reflections in the river was considered erroneous labeling. Also, the identification of a known building (such as the Fulton Building) as a "miscellaneous building" was considered an error since the system failed to obtain the most accurate label. A good measure of the recognition quality is the system's ability to identify the viewpoint. In 80% of the images key objects were labeled that demonstrated a recognition of the correct view. For example, identification of the correct rivers, bridges, and roads indicates an understanding of the viewpoint whereas identification of buildings does not since they are discernible from all angles.

Another proposed development is the use of knowledge hierarchies for improved labeling. As a first-pass, the system will extract viewpoint information from one run of ARGOS and then construct view-specific knowledge for more accurate re-evaluation of the image.

IMAGE FEATURE ANALYSIS

Kanade is developing techniques for identifying task-independent knowledge sources. One of the recent developments in this area is a generalization of Huffman-Clowes-Waltz techniques for line labeling (Kanade, 1978b) in this workshop. This generalization is more flexible than the conventional trihedral world, where solid objects are the

basic components, since it accepts a larger class of drawings which are usually obtained by processing real-world images. Various local cues extracted from the image (e.g. edge cross-section profile, collinearity, etc.) can be exploited to constrain the possible interpretations. This method allows one to incorporate structural (junction types and connections), geometrical (line direction and collinearity), and spectral (color and intensity characteristics at the edges) information for determining the 3-D configuration of an object in an image. A more detailed description of this work is given in Kanade (1978a).

INTERACTIVE AIDS

We have continued our work on the MIDAS sensor database. We have concentrated our efforts on bringing up an image browsing facility on MIPS. This facility is operational and has the capability of displaying color or color mapped images in a variety of resolutions. A user can quickly browse through a collection of images and zoom in on particular areas of interest by displaying windows of the image in different levels of resolution. Using symbolic region descriptions (McKeown and Reddy, 1977) it is possible to symbolically address any portion of the image. A simple query system allows users to point at the display and retrieve pre-segmented region information.

We plan to expand the utility of this system by increasing the variety of images available for these types of queries (currently 20) and investigating methods for responding to queries where responses must be generated from the signal data. Preliminary work has begun in acquiring map data and associated aerial photography for our Washington D.C. task. We have begun work on evaluating map representations and their suitability for photo interpretation tasks.

ARCHITECTURES FOR IMAGE PROCESSING

We are beginning work on algorithm decomposition for parallel processors, an area in which we are fortunate to have two working systems: Cm* and C.mmp. Cm* is an example of an asynchronous parallel processors organized as a network of clusters of processors (Swan, 1976). Currently there are 10 LSI-11 microprocessors in the system. Over the next two years we expect to have a 50 processor system and evaluate its effectiveness in a real-time image understanding task. This organization provides significant flexibility, allowing each processor to execute different operations and perform different computations. One important question is how do we organize algorithms to effectively use asynchronous parallelism of this type? Preliminary explorations indicate that by careful organization in a parallel pipeline, modular algorithm decomposition will permit the full realization of the parallelism potential.

Our joint research effort with CDC continues to be the development of a low cost high-speed processor element with special functional units for image and symbol manipulation operations. We currently believe that the processor should be ready for testing and validation within the next year. The addition of these processors to the MIPS machine should greatly facilitate much of our low-level processing operations.

With the availability of writable control stores, it has become possible to modify and add to the instructional set definitions of current computer systems. Experiments performed on our PDP-11/40E processors indicate that a 10 to 30 percent improvement in performance can be expected for certain image processing tasks. We are currently engaged in a study of the design of special instruction sets for image processing applications. We should expect to have architectures which execute as primitive instructions certain high level operations on images. An analysis of the costs associated with certain operations, including frequency and locality of memory accesses, parameter passing schemes, and arithmetic complexity is underway.

CONCLUSION

While the primary emphasis continues to be in effective use of knowledge in the image interpretation process, the research at CMU is tempered by the realization that we must also pay adequate attention to other relevant aspects such as computer architecture, software design, image databases, performance analysis and perceptual psychology. We continue to have modest efforts in each of these areas.

REFERENCES

Kanade, T. (1978a). "Origami Understanding" CMU Technical Report, Department of Computer Science. 1978. (in preparation).

Kanade, T. (1978b). "Task Independent Aspects of Image Understanding", in this volume.

McKeown, D. M. and Reddy, D. R. (1977). "A Hierarchical Symbolic Representation for an Image Database" Proceeding of *IEEE Workshop on Picture Data Description and Management*, April, 1977.

Rubin, S. M. (1978). "The ARGOS Image Understanding System", Ph.D Thesis Department of Computer Science, Carnegie-Mellon University.

Swan, R. J. *et al.* (1976). "Cm*: A Modular, Multi-Microprocessor," *AFIPS Conference Proceeding*. Vol. 46, 1977 National Computer Conference, pp. 645-655.

AUTOMATIC IMAGE RECOGNITION SYSTEM

Program Status, March 1978

R. Larson

HONEYWELL INC.
Systems and Research
Minneapolis, Minnesota 55413

This program is focused on tactical applications of image understanding. The constraints on a tactical system, autonomous, mission directed, real time operation in an unknown environment, place some severe limits on the methods that can be used and tend to make the solutions problem dependent. Thus, while tactical solutions can be applied to intelligence and strategic problems, the converse is not generally true.

The program goal is to develop and simulate algorithms needed in the system diagram shown on page 135 of the September 1977 IU Workshop proceedings. Our recent work has been on the target recognition part of the system, in particular the secondary screening of man-made objects and syntactic classification of large images.

CURRENT IMAGE UNDERSTANDING EFFORT

Large Image Classifier - We define a large image as one that is observed at sufficiently high resolution that object structure can be seen. Statistical recognition is very difficult for large images of three dimensional objects because of the large number of viewing angles that must be treated. We have, therefore, concentrated on methods that refer the analysis back to the structure of the object in either a numerical or a symbolic way. The numerical methods were studied at Purdue and reported by them. The symbolic methods are being studied by Honeywell using the prototype similarity transformation. Both approaches begin by having a potential targets' position in the frame and its approximate size designated to the algorithm (the results of the man-made object detection and secondary screening). The algorithm then segments the part of the frame near the designated position into target and non-target pixels and extracts features or primatives from the target part. In the Honeywell approach the subimage is transformed into low level symbols and the segmentation is done on the symbolic image. The target image components are then obtained by using prototype similarity again with finer resolution. Work on recognizing the components and, from them, the target is just beginning. Statistical classification will be investigated for recognizing the unresolved components and syntactic methods will be used for object recognition.

Interframe Analysis - In tactical imagery we can expect the unobscured, high contrast target image to be a rare event. The target objects will be moving across a background of varying intensity and background objects will frequently obscure parts of the target. Even when we are close enough to the target to be able to resolve its component parts, the contrast and obscuration effects will make it difficult or impossible to obtain a complete target image from any single frame. It is, therefore, necessary to be able to track objects from frame to frame and to construct a composite image from the separate frame approximations. In this we are also studying both numerical and symbolic methods. At the last workshop we reported on interframe tracking experiments done by passing prototypes from frame to frame. The paper by Panda at this workshop reports our initial numerical results. We are studying ways to use the resulting data sequence to increase the recognition accuracy.

Configuration Analysis - This task deals with the final step in the image understanding problem; obtaining a description of the scene from the list of recognized objects. Scene description in the sense of describing a scene in terms of recognized target and background objects and their relative locations is important in a number of multi-warhead autonomous system concepts. It is also significant in intelligence, navigation and certain terminal homing concepts. For the system we have defined the scene description/configuration analysis is to be used only to identify complex targets that are composed of a number of individual, recognizable objects (e.g. a convoy of vehicles or a power plant). Two problems that must be solved in configuration analysis are 1) how to represent the information, and 2) how to generate the desired kind of description. A review of existing methods has led us to select a rule-based network (production rules linked as a network) as the data representation and a bottom up analyser as the control structure for generating descriptions. A part of the rationale for these choices is that they will allow the representation of both relationships and properties in the same structure and we feel that this is a necessary capability.

Secondary Screening - Target classification is preceded by a number of data bandwidth reduction steps that also serve to direct the attention of the classification activities. Target cueing is done upon image characteristics. In our system we follow the cueing by a screening based upon object dimensions. These are deduced from the image, the sensor resolution and the sensor to object range. The dimensions are compared with known target dimensions and accepted objects are passed to the appropriate classifier. This function has been implemented as a software modification to the Honeywell Autoscreener and tested against FLIR imagery recorded in television format. As expected, the secondary screening decreases the number of potential targets that must be processed by the classifier. We have found that rejection of targets by the secondary screening is determined by the accuracy of our range estimate, which, in our experiments, was determined by estimating the depression angle of the sensor.

Autothreshold - The Autothreshold is an image segmentation method based on thresholding an image relative to an estimate of the scan line intensities derived from the previous scan lines. The method adapts well to the varying intensities found in tactical imagery (both interframe and intraframe variations are significant), and the method is readily implemented in either analog or digital hardware. Since the last workshop report a similar method of background estimation has been incorporated in an image enhancement circuit using discrete CCD's.

IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

T. S. Huang and K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

OVERVIEW

The objective of our research is to achieve better understanding of image structure and to improve the capability of image processing systems to extract information from imagery and to convey that information in a useful form. The results of this research are expected to provide the basis for technology development relative to military applications of machine extraction of information from aircraft and satellite imagery.

We are carrying out basic and applied research in the following four overlapping areas: preprocessing, image segmentation, image attributes (especially texture and shape analysis), and image structural analysis. The long-range goal of our research is to find good symbolic representations for images, and techniques for transforming raw image data into such representations. In the immediate future, the emphasis of our research will be on one or more of the following: 1) combining syntactic methods with statistical pattern classification techniques. 2) Understanding moving images. 3) Efficient computer implementation of image understanding algorithms.

SUMMARY OF RESEARCH PROGRESS

Preprocessing (Huang, O'Conner, Yang)

We have initiated a basic research project in nonlinear image enhancement techniques. Of particular interest is the problem of reducing noise in images without blurring the sharp edges contained therein. Our approach is to decompose the image into several components in such a way that the noise characteristics in the components are more amenable to nonlinear filtering methods. One particular class of nonlinear techniques under study is median filtering and its extensions. A fast two-dimensional median filtering algorithm has been developed and programmed on our PDP 11/45 computer. It is several orders of magnitude faster than the most efficient sorting methods.

Another area we are investigating is the comparison of three phase unwrapping techniques in regard to estimating the point spread function of image degrading systems. Preliminary results indicate that they comlement each other and perhaps should be combined in some manner.

Picture Source Coding (Mitchell, Delp, Carlton)

We have been tasked by Rome Air Development Center to investigate compression techniques for image transmission systems where a human analyst is the ultimate receiver of the picture. This has led to a comparison of many existing techniques and the development of a new spatial coding technique which is highly matched to the human observer, is simple to implement, and is comparable to much more computationally intensive coding methods.

Binary Array Processing and Image Registration (Reeves)

Research has been conducted in two main areas. First, a computer system for testing image processing algorithms has been implemented. Then this system has been used to test a novel scheme for image registration. A general purpose, Fortran based, array processing system, APS, has been implemented on the PDP-11 computer.

The system is designed to simplify the programming and testing of image processing algorithms. The data structure for image uses a bit-plane format rather than the more conventional sequential file. To assist with the processing of large arrays, APS features include dynamic array storage allocation and a virtual memory for arrays.

This system was originally designed to simulate a binary array processor called BASE. As a consequence of this, programs written in APS are well structured for parallel array processing.

APS is written in Fortran for portability but contains some assembly code sections. The present version runs on a PDP 11 computer under the UNIX operating system. A library of image processing subroutines is being developed which is completely portable with respect to any machine which runs APS.

The system has been coupled with a high level language interpreter so that both high level interactive programming and efficient execution can be achieved.

A scheme has been developed for the rapid registration of a sequence of images. This scheme is suitable for applications involving a FLIR or a conventional TV system. Each image is converted into a binary feature image. Feature images may be rapidly registered and also any movements of significant objects within the image can be detected.

An image is first processed to remove artifacts caused by the imaging equipment. In the case of FLIR images a 3 bit wide median filter is first applied along each line to remove noise pixels and then adjacent lines are added together to remove banding effects. The binary feature image is then computed from the preprocessed image according to the equation

$$F = \begin{cases} 1, & \text{if } (P\text{-}MIN) > (MAX\text{-}P) \\ 0, & \text{otherwise} \end{cases}$$

Where P is the pixel value, MAX is the local maximum and MIN is the local minimum. The size of the area over which the local maximum and minimum are computed is a system variable.

This scheme has been tested on the APS system. All stages of processing may be rapidly executed on a parallel binary array processor.

Image Segmentation by Edge Detection (Huang, Salahi, Tang)

Image segmentation by edge detection consists of two steps. First, edge points are detected. Then, these edge points are connected to form curves. We have developed a syntax-semantics guided technique for accomplishing the second step. The resulting edge strings are generally disconnected. Currently, we are investigating target classification techniques which do not require closed boundaries.

Use of Fourier Boundary Descriptors to Classify Three-Dimensional Aircrafts (Wintz, Wallace)

As described in our last progress report, our work recently has dealt with the application of Fourier descriptors to recognition of three-dimensional aircraft recognition. Since our last report, we have achieved results comparable to those of Dudani [1] using a projection density nine times lower than he used, and considering a much larger sector of three-space. The property of frequency domain interpolation of FDs was exploited in our algorithm, enabling the reduction in projection density.

This algorithm is of considerable interest in its own right, but it also suggests a possible approach to the problem of recognition of partial shapes extracted from photographic area. One weakness of FDs, as presently used by our algorithms is the fact that the entire object must be roughly extracted from the picture for classification to be successful. An aircraft with one wing missing due to shadow will not have similar Fourier descriptors to one which is intact. This is because the FD is a frequency domain expansin of the entire outline of the shape being analyzed.

Our three-dimensional algorithm defines a natural projection space of two-dimensional projections taken at successive rotations about the x and y axes. We can define a space of FDs parameterized by the equation of a line which may cut off part of the desired object due to shadow, noise, or any other obstacle to obtaining the complete outline. While a straight line might not exactly model the division of the object, the noise-rejection properties of our whole procedure should result in an algorithm which can detect part of an object even when the division is only very roughly straight. A more quantitative statement depends on the ratio of perimeter belonging to the object to perimeter belonging to the dividing "line." The interpolation algorithm would be applicable to this situation, but a first practical implementation of this would not include three-dimensional capabilities.

Since our Fourier descriptor algorithms have been successfully tested, we plan to apply them to more real data. We also plan to continue theoretical investigation of FD theory in an attempt to both improve present methods and define their limits.

FLIR Image Segmentation and Target Tracking (Mitchell, Carlton, Ward)

During the past months, research in texture and segmentation has been advancing especially in the applications to FLIR imagery target recognition and real time video target tracking. We have been studying methods of automating the texture extremal thresholds to make the method more adaptive. Our use of texture measures is centering in two primary areas: (1) texture edge detection and texture region growing to segment an object from its background and (2) classification of background regions for use in the higher-level global recognition system. The data sets we are using primarily are the FLIR large target data set from Honeywell (120 images with identified tactical targets) and the White Sands Missile Range TV data set (20 digitized images--an additional 150 images are soon to be added).

The method of projections is being investigated for structure analysis of the segmented images as well as boundary descriptions for tracking the changing shape of an object as it moves and as the sensor moves.

Syntactic Algorithms for Image Segmentation and a Special Computer Architecture for Image Processing (K.S. Fu and J. Keng)

Several efficient algorithms for image recognition and segmentation and a new computer architecture for image processing are proposed. The algorithms are "syntactic" in that they perform structural or spatial analysis rather than statistical analysis, and a "grammar" is inferred for describing the structures of patterns in an image. Depending on the requirements of the problem, an appropriate grammatical approach is used by the syntactic algorithm.

A finite-state string grammar is applied to the image recognition of highways, rivers, bridges, and commercial/industrial areas from LANDSAT images. There are two major methods in the string grammar approach for image recognition; namely, the syntax-directed method and suntax-controlled method. For the syntax-directed method, syntactic analysis is performed by a template matching which is directed by the syntactic rules. For the syntax-controlled method an automaton which is direct-

ly controlled by the syntactic rules is used for the syntactic analysis.

A tree grammar is applied to the image segmentation of terrain and tactical targets from LANDSAT and infrared images respectively. The tree grammar approach utilizes a tree automaton to extract the boundaries of the homogeneous region segments of the image. The homogeneity of the region segment is obtained through texture feature measurements of the image.

The computer architecture proposed is a special purpose system in that it can perform an image processing task on several picture-points of an image at the same time, and thus takes advantage of the fact that image processing tasks usually exhibit "parallelism." This architecture uses a distributed computing approach. Two major features are the reconfigurable capability, and the method of computer exploitation of task parallelism. Finally, a parallel parsing scheme for tree grammar is used to demonstrate the higher efficiency of the proposed computer architecture than the conventional parsing scheme.

Syntactic Shape Recognition Using Attributed Grammar (K.S. Fu and K.C. You)

Syntactic method has been studied in pattern recognition and image processing [2], our approach attempts to develop a more general method for shape recognition. By shape, we mean the outer boundary of the two dimensional image of an object. Being the most intelligent recognizer, humans recognize the shape by analyzing its structure by grammatical rules and the local details by primitives.

Four attributes, or feature values, are proposed to describe an open curve segment, and the angle between two consecutive curve segments is used as the attribute to describe the connection. Any connection angle is called an angle primitive, while a curve primitive is an open curve segment with a curvature function which is either positive or negative throughout the curve segment. The four attributes are $\vec{C}$, L, A and S. $\vec{C}$, L are the vector from one end to the other, and length of the curve respectively.

$$A = \int_0 f(1)\, dl, \quad S = \int_0 \left(\int_0^s f(1)\, dl - \frac{A}{2}\right) ds.$$

Where f(1) is the curvature function of length L from one end. That is, A is the total angle of the curve, while S somehow indicates the symmetry of the curve. The four attributes are defiined as the C-descriptor of a curve segment (not of a curve primitive only) and the connection angle the A-descriptor of an angle primitive.

The C-descriptor, after a transformation

$$T = (\vec{C}, L, A, A) = (|\vec{C}|/L, A, S/L, L/L_0),$$

in which L_0 is the total length of the shape, and the A-descriptor are theoretically invariant under rotation, translation and scaling. Unfortunately, the digitization introduces noise into the picture after any of the above operations. The effect is investigated. Some interesting properties and efficient computation in discrete case of the descriptors are studied.

The shape, after proper segmentation, can be described by an attributed grammar [3], in which each primitive or nonterminal has a descriptor as its attributes. The descriptors are computed when the recognition is performed.

The primitive extraction from noisy pattern is usually very difficult. The grammatical information and looking ahead techniques can be used to optimize the extraction. The Earley's algorithm is modified to embody the primitive extraction into the parse list generation, so that the input of the algorithm is a vector chain instead of a primitive string.

The shape grammar can be converted from a context-free grammar to a finite-state grammar, which is more efficient in processing. The primitive extraction can also be embedded in the corresponding finite-state recognizer. Recursive expressions for computing the descriptors are developed to speed up the process. Grammatical inference directly from the noisy patterns can also be implemented automatically or interactively.

REFERENCES

[1] S.A. Dudani, et. al., "Aircraft Identification by Moment Invariants," IEEE Trans. on Computers, Vol. C-26, pp. 39-46, January 1977.

[2] K.S. Fu, Syntactic Method in Pattern Recognition, Academic Press, 1974.

[3] P.M. Lewis, II, D.J. Rosenkrantz, and R.E. Stearns, Compiler Design Theory, Prentice-Hall, 1976.

PROGRESS AT THE ROCHESTER IMAGE UNDERSTANDING PROJECT

C. M. Brown
J. A. Feldman

The University of Rochester
Rochester, New York 14627

1. Model Refinement

1.1. Procedural Description

One important goal of the Rochester Vision Project is to investigate a generalized form of procedural invocation in which an executive procedure chooses worker procedures to perform a job not just on the basis of input/ output behavior (as traditional pattern- directed invocation does), but also taking into account cost/ benefit estimates and perhaps other information as well. This scheme is motivated by the desire to have the advantages of declarative knowledge about what is doable (the descriptions) along with the advantages of procedural knowledge about how to do it (the workers). The declarative, descriptive component will allow conviences such as the modular addition of procedural knowledge. The main research issue is to decide what exactly needs to be known about worker procedures, and how to express that in a useful and uniform manner. The most recent and presently contemplated work at Rochester explores aspects of these issues (e.g. Lantz, Ballard, and Brown, 1978).

1.2. Decision Theory

The use of decision theory not only as an abstract model of intelligent perception but as a practical tool to maximize computational benefit/ cost is being investigated in the context of procedural invocation. This work continues in the tradition of Bolles, Sproull, and Garvey, and ultimately we hope to extend some of their results to deal with formal problems that more closely approximate the sorts of vision problems encountered in our particular applications. Ballard (see Section 2) uses decision theory techniques to choose the most economical method (assuring adequate accuracy) of locating anatomical structures in large-format images.

2. Applications in Biomedicine

The model-directed finding of ribs in chest radiographs ([Ballard, 1978]) provides an illustration of the use of the Rochester Vision System, incorporating procedure description, utility measures, and tops-down, model-directed perception. The object here is to cope with large amounts of possibly low-quality data without undue processing time by depending on a declarative model of anatomical structures, described procedural knowledge about how to locate them, and an executive which uses decision theory to control the image- understanding process.

A novel and uniform method of describing arbitrary functions on the unit sphere (which define "museum-viewable" volumes) is under investigation, with immediate application to anatomical structures [Schudy 1978]. The idea is related to the well- known Fourier descriptions of two- dimensional shape. Volumes are modelled and described as the leading coefficients in certain spherical harmonic expansions of the volume functions. This method also allows least squared error fitting of volumes in coefficient space, which interfaces nicely with routines which locate the three- dimensional boundaries of volumes in image data.

3. Application in Aerial Image Analysis

The three-level organization of image analysis (strategist, executive, worker) and a further exploration of useful procedural description mechanisms are the objects of study in automatic photo- interpretation work ([Lantz 1978]). The object is to use the sorts of knowledge- based inferencing used by skilled photointerpreters, along with models inspired by photointerpretation keys for identifying small industries, to do reliable and flexible

identification of a few types of small industrial installations. Imagery has been acquired from a Rochester, N.Y. mapping firm and from RADC in Rome, N.Y. We plan to digitize the images in cooperation with other DARPA contractors (Maryland or USC). In the meantime, modelling issues are being addressed.

4. Fast Display of Certain Polyhedra

The descriptions of 3-D vector data histograms mentioned in previous reports are only an instance of a general class of polyhedra for which unusually quick solutions exist to the hidden line/surface problem. In the last six months, the conditions guaranteeing quick displayability have become understood, and display programs written to use the resulting algorithms ([Brown 1978]). Also recently the original statistical motivation for the work has received more attention ([Wellner 1978]).

5. Component Building

5.1. Hardware

The Grinnell GMR-26 display device is on site and DMA-interfaced to the second (Vision) Eclipse computer. 32K of core has been added to the Vision Eclipse, which is also used for research in distributed computing (see Section 5.2). The original 80MB disk has been replaced with a 300MB one, and another 300MB disk is on order, to arrive in April 1978. We are acquiring terminals and investigating how to meet our everyday computing needs by commercial, home-built, or combination intelligent terminal systems. Acquisition of a frame-rate TV-based digitizing device is still proceeding. Construction of a fast (50KB) link to the PDP-KL10 is nearly complete.

5.2. Software

Advanced system software support is now used routinely, and more is under development. Communications protocols and distributed computing packages ([Rovner 1978, Feldman 1978, Sheininger and Sabbah 1978, Selfridge 1978, Sloan 1978]) have been developed to allow access to the GMR-26 through the local ALTO computers or the remote PDP-10, to achieve reliable transmission between distributed processes, to produce graphics and halftone images on ALTO screens from the PDP-10, and to allow file transfer and telnet to the Arpanet. The IPCF in the TOPS-10 operating system is the basis for communication between PDP-10 jobs, and these jobs may now create RIG messages and send them to the local operating system for disposition. At Rochester, the RIG message is the lingua franca that allows processes on remote machines to command the GMR-26, perform file manipulations, etc. While at SRI International for the summer, K. Lantz wrote systems code for the multiple process HAWKEYE system [Barrow et al. 1977]. Some student projects in our Computer Vision course are aimed at producing useful system software for vision, and the common departmental interest in distributed computing assures that new and co-operative efforts using the distributed computation and communications packages will be launched frequently. A comprehensive library of vision routines ([Sloan 1977-78]) has been developed, centralized, documented, and incorporated into the NEXUS system. They allow interactive users a wide range of image-processing and display (graphics, halftone, color and B&W TV) capabilities.

6. Motion Understanding

Understanding motion pictures has always presented an unusually difficult problem to computer vision efforts. The compelling gestalt induced in humans by moving objects is not well understood, and so there is little leverage on the immediate problems resulting from the large mass of data in multi- frame images. We are hoping to make progress first on a pared-down version of the problem which nevertheless offers an interesting set of perceptual phenomena to model. The domain is multi- frame images of animal motion; initial research is being carried out on sequential images of points of light attached to joints. This data can give humans a strong perception of coherent motion, and present work is aimed at understanding how we correctly identify points (about 13 in all in present data) from frame to frame, and how we segment the resulting moving points into meaningful body parts. Ultimately, the results will be applied to multi-frame grey-scale images. Data presently comes from a program which simulates a range of human walking motion in 3-D. The program is a useful theoretical tool, since it allows direct access (not mediated by vision) to movement parameters, point locations, etc. It is also a useful psychological research tool, since with it one can inexpensively investigate limits in human performance.

7. Programming Language Development

The Smart Compiler and Distributed Computation research groups are cooperating on a language for research into both these fields ([Ball 1978]). It will contain the ideas of PLITS, together with improvements and extensions gleaned from the SAIL-PLITS implementations of the past. There are several separate ways in which the programming language developments are affecting Image Understanding research in our laboratory and elsewhere [Feldman & Williams 1977]. An overview of this work is presented in the companion paper in this volume [Feldman 1978].

REFERENCES

Ball, J.E., et al., ZENO: a language for smart compiler research, Internal Memo, Computer Science Department, University of Rochester, (in preparation) 1978.

Ballard, D.H., Model-directed detection of ribs in chest radiographs, TR11, Computer Science Department, University of Rochester, March 1978.

Barrow, H.G., et al., Interactive aids for cartography and photo interpretation, Semiannual Technical Report, Artificial Intelligence Center, SRI International, November 1977.

Brown, C.M., Fast display of certain museum-viewable polyhedra, TR23, Computer Science Department, University of Rochester, March 1978.

Feldman, J.A., Synchronizing distant cooperating processes, TR26, Computer Science Department, University of Rochester, October, 1977.

Feldman, J.A., Systems support for advanced image understanding. DARPA Semiannual Technical Report, May 1978.

Feldman, J.A., and Williams, G. Some comments on datatypes, TR28, Computer Science Department, University of Rochester, December 1977.

Lantz, K.A., Procedural knowledge and control in a model - driven vision system, Thesis proposal, University of Rochester, February 1978.

Lantz, K.A., Ballard, D.H., and Brown, C.M. General invocation through procedure descriptions: two applications in image analysis, to appear in 22nd International Symposium of the Society of Photo-optical Instrumentation Engineers, San Diego, CA., August 1978.

Rashid, R.F. Motion understanding, Thesis proposal, University of Rochester, in preparation 1978.

Rovner, P.D. Flow control and reliable transmission in a system for distributed computing, TR22, Computer Science Department, University of Rochester, October 1977.

Rovner, P.D. Automatic representation selection for associative data structures, to appear in Proceedings of the National Computer Conference, Anaheim, CA., June 1978. Sabbah, D. Image calibration, Internal Memo, Computer Science Department, University of Rochester, in preparation 1978.

Scheininger, U., and Sabbah, D., The display process, Internal Memo, Computer Science Department , University of Rochester, December 1977.

Schudy, R., A model for echocardiography, TR12, Computer Science Department, University of Rochester, (in preparation) 1978.

Selfridge, P., Name - value pairs in the Rochester vision header. Internal Memo, Computer Science Department, University of Rochester, January 1978.

Sloan, K.R., Rochester vision library documentation, Internal Memos, Computer Science Department, University of Rochester, 1977 - 1978.

Wellner, J.A., Two-sample tests for a class of distributions on the sphere, submitted for publication, February 1978.

SPATIAL UNDERSTANDING

Thomas O. Binford

Artificial Intelligence Laboratory, Computer Science Department
Stanford University, Stanford, California 94305

Abstract

This report discusses research in two areas. The first is spatial interpretation of stereo images and use of context in stereo mapping. The second is the design and construction of a model-based system for finding airfields and other objects from generic models.

Introduction

A major objective of this research is to solve scientific problems encountered in using stereo vision and motion parallax for photointerpretation, mapping, and guidance. Stereo ranging is attractive because it is passive, has high depth accuracy, high spatial resolution, and makes use of images from visible, SAR, and other well-developed sensors. Several systems have partial success with automated stereo terrain mapping. They have problems around buildings (surface discontinuities) and on surfaces which are uniform or have repetitive markings (the ambiguity problem). The chief problems to be solved are: automating the process of matching corresponding parts of images, particularly at surface discontinuities; resolving ambiguities by using more global correspondence; designing algorithms and machine architectures to meet time objectives.

We have taken the approach of building spatial models of surfaces in order to make use of a priori knowledge about objects, and to construct a consistent context within the scene. Knowledge from outside the scene and from within the scene are used to reduce ambiguity. We have used both feature correspondence and small area correlation. The two are complementary. Edge correspondence is useful at discontinuities of uniform surfaces. Area correlation is useful with textured surfaces. For identification, we match three-dimensional structures, rather than two-dimensional images. To make stereo mapping fast we have developed coarse-to-fine search strategies and utilized edge-based matching. These are combined with successive approximation modeling, which concentrates effort at any stage on large unmatched areas.

A second research objective is to build a system which locates airfields in aerial photographs. It must do this from a dialog with a PI and a set of examples. The system should be generic. That is, the same system will be used to locate oil tanks, based on another dialog and a set of examples. The same system will be used for aircraft and vehicle identification. An important consideration in a dialog is the language in which it will be carried out. In this case, a language common to users and to our vision system was chosen, a language of object models.

Stereo Vision

Arnold [1] describes results obtained with photos of San Francisco Airport, an apartment building, and a parking lot. The system requires one minute of machine time to make a depth map of edges of surfaces. The edge map appears easily adequate for identification. Edge maps are relatively continuous with few errors; they are improved near corners and along edges which are nearly parallel to the stereo axis. The system has been rebuilt, with memory management to work with very large images, and is now being tested. Some of the weaknesses of current edge operators show up under the close scrutiny of image matching.

This research aims at high resolution of surface boundaries to make measurement of dimensions and angles. It is about a factor of 10 more accurate for such measurements than Gennery's system [2]. It is thought effective with thin objects such as poles, although no examples have been demonstrated. An essential part of the research is the use of context in matching. The system currently uses local context of edge continuity, and the context of the ground plane. The system is being extended to use context of locally planar surfaces, with successive approximation modeling. The addition of greater context is expected to produce effective depth mapping.

A model for stereo vision is emerging. The model is based on surface interpretation of edge and area correspondence, with a coarse-to-fine search strategy, and successive approximation of surface models.

Model-Based Vision

In order to make a system which an expert PI can use to locate airfields, it is necessary to provide means for the expert to express the task and his knowledge. This knowledge is a combination of geometric and symbolic. We have taken the approach of building a high level language for object modeling to express spatial structures and relations [3].

The system uses that knowledge by building a structure for what it expects to see in the picture. Some of the expectation is generic, i.e. widely applicable, some specific to the task at hand. It determines symbolic observables and relations and links them to their interpretations in the object models.

A model-matching program uses multi-level relaxation in the form of coarse matching and detailed matching.

The system can be driven in the other direction. Structures from the picture can be mapped to generalized cones and three-dimensional object interpretations. It can thus build scene descriptions guided by object knowledge. This level of generality is very promising.

The system is largely not probabilistic. It does not have distributions of expected pictures or objects, but it does have a partial ordering among perceptual operations in terms of expected cost and effectiveness. It has models of what it expects to find, but not models of the rest of the image. Thus, it does not have a good way of distinguishing desired objects based on very simple discriminations such as color. Instead, to make effective selection of initial candidates it must use local shape and context. To match, it must require strong reinforcing structural evidence, not discount known alternatives. Only in this way can it function in a complex visual environment.

The system is partially implemented. We expect to use it to identify aircraft from stereo maps produced by the edge-based stereo system.

References

1. Arnold, R.D.; "Local Context in Matching Edges for Stereo Vision"; In these proceedings.
2. Gennery, D.B.; "A Stereo Vision System"; Proceedings ARPA Image Understanding Workshop, October 1977.
3. Brooks, R.A., Greiner, R., Binford, T.O.; "A Model-Based Vision System"; In these proceedings.

AN EXPERT SYSTEM FOR DETECTING AND INTERPRETING ROAD EVENTS DEPICTED IN AERIAL IMAGERY

H.G. Barrow
M.A. Fischler
SRI International
Menlo Park, California

ABSTRACT

This paper presents an overview of SRI International's effort to construct a "Road Expert" whose purpose is to monitor and interpret road events in aerial imagery. Goals, approach, and the current state of this research are described.

INTRODUCTION

Research in Image Understanding at SRI International was initiated in 1975 to investigate ways in which diverse sources of knowledge might be brought to bear on the problem of analyzing and interpreting images. The initial phase of research was exploratory in nature, and identified various means for exploiting knowledge in processing aerial photographs for such military applications as cartography, intelligence, weapon guidance, and targeting. A key concept is the use of a generalized digital map to guide the process of image analysis.

The results of this earlier work were integrated in an interactive computer system called "Hawkeye" (see Ref. 1). Research has now focused on a specific task domain: road monitoring. The following sections of this report present an overview of this new effort.

OBJECTIVE

The primary objective in this research is to build a computer system that "understands" the nature of roads and road events. It should be capable of performing such tasks as:

(a) Finding roads in aerial imagery

(b) Distinguishing vehicles on roads from shadows, signposts, road markings, etc.

(c) Comparing multiple images and symbolic information pertaining to the same road segment, and deciding if significant changes have occurred.

It should be capable of performing the above tasks even when the roads are partially occluded by clouds or terrain features, or viewed from arbitrary angles and distances, or pass through a variety of terrains.

APPROACH

To achieve the above capabilities, we are developing two "expert" subsystems: the "Road Expert" and the "Vehicle Expert." The Road Expert knows mainly about roads, how to find them (in imagery), and what things belong on them. It works at low to intermediate resolution (say from 1 to 20 feet of ground distance per image pixel) and has the ability to distinguish vehicles from other road detail. The Vehicle Expert works on higher-resolution imagery and can identify vehicles as to type. We are concentrating our initial efforts on the Road Expert, and therefore will limit our discussion to this component of our system.

Among the specific tasks to be performed by the Road Expert are the following:

(1) Place the image into correspondence with the map data base

(2) Determine the precise location of known roads in the image

(3) Determine the visibility of the located road segments

(4) Mark the road center-line and lane boundaries

(5) Detect anomalous regions on and along the road pavement

(6) Determine which anomalies are potential vehicles.

The image/map correspondence task will be accomplished primarily by using roads as landmarks; thus, Tasks 1 through 3 will interact strongly with each other. These tasks will be performed at approximately 20 feet/pixel resolution so that a reasonably wide field of view (10 to 100 square miles) can be processed at one time.

Having located visible portions of roads, individual sections will be selected for detailed analysis. Increasing resolution to approximately 1-3 feet/pixel, the road center-line and lane boundaries will be found starting with the initial estimate obtained in the low-resolution step. We will then detect anomalous regions on and along the road pavement, and finally decide which of these regions are vehicles. Since road anomalies will cause problems in tracking a nominally homogeneous road surface, Tasks 4 through 6 will be integrated to some extent.

The above tasks will be supported by information about road condition and general structure from a symbolic data base. For example, if prior photographic coverage of the area being analyzed is available, the problem of anomaly classification can be simplified by determining if a similarily shaped anomaly was found in the same general location over some extended period of time. Additional examples of how data-base knowledge and stored models can aid in the analysis process include: the use of time of day in discriminating shadows from objects of interest; the general shape and width of the road (as obtained from a map) to aid in road tracking; and the expected size, shape, and road orientation of potential vehicles.

A central theme of this effort is to consider Roads as a knowledge domain. In particular, we plan to address the question of how a-priori knowledge can be directly invoked by the image processing modules (what type of knowledge; how should it be represented; what are mechanisms for its use). To achieve our goal of building a very-high-performance system, we plan to develop explicit models of the image structures we will be dealing with and, additionally, models of the decision procedures embedded in the image-processing algorithms so that the algorithms can evaluate their own performance. Finally, we must develop an overall control structure, which will be concerned with the problems of coordinating analysis across a number of levels of resolution, and with integrating multisource information.

PROGRESS

Working programs exist that are capable of performing each of the major tasks to be performed by the Road Expert; however, these programs are low-level in the sense that they still cannot communicate with each other, or modify their performance based on context or self-evaluation. In almost all cases, the level of performance is expected to improve substantially as we integrate the individual modules and modify them to accept data-base support.

We are currently placing major emphasis on Tasks 3 through 6, and some of this work is described in a companion paper by Lynn Quam (Ref. 2). Using a road model that assumes segments exhibiting relatively smooth/slow changes in direction and also in the intensity profile normal to road direction, we have been able to achieve surprisingly robust the performance in tracking the road center-line. In many cases, roads that have almost no discernible contrast at their edges can be reliably followed.

In order to support our experimental work, we have acquired multiple photographic coverage of five distinct sites scattered around the San Francisco Bay Area. This imagery (most of it still to be scanned) shows road detail at the resolutions mentioned earlier--i.e., 1 to 20 feet of ground distance per image pixel.

CONCLUDING COMMENTS

We see the military relevance of our work extending well beyond the specific road-monitoring scenario presented above. In particular, a Road Expert can be applied to such problems as

(1) Intelligence: monitoring roads for movement of military forces

(2) Weapon Guidance: use of roads as landmarks for "Map-Matching" systems

(3) Targeting: detection of vehicles for interdiction of road traffic

(4) Cartography: compilation and updating of maps with respect to roads and other linear features.

In accord with our generalized view of the applicability of the Road Expert we are constructing, we will attempt to achieve a level of performance and understanding in each of the functional tasks that far exceeds that required for dealing with the road monitoring scenario alone.

REFERENCES

1. H.G. Barrow, et al., "Interactive Aids for Cartography and Photo Interpretation: Progress Report, October 1977," PROCEEDINGS: IMAGE UNDERSTANDING WORKSHOP, October 1977, pp 111-127

2. L. Quam "Road Tracking and Anomaly Detection," PROCEEDINGS: IMAGE UNDERSTANDING WORKSHOP", May 1978. In Press.

USCIPI SIX MONTHS OVERVIEW

Harry C. Andrews

Image Processing Institute
University of Southern California
Los Angeles, California 90007

The past six months have been quite productive on a variety of research and development fronts. The image understanding projects are maturing with symbolic matching, structure location, edge fitting, stochastic texture analysis and SVD feature selection, all being reported upon in some detail. The image processing projects present both new and concluding projects. New projects include double phase binary computer generated holograms and turntable radar imaging via coherent multi-frequency radar return processing. Older projects resulting in successful theoretic and experimental work include a posteriori restoration and perceptual model color image coding. Our on-going smart sensor project is expanding rapidly with old circuits being driven at near real time TV rates and new circuits being designed for 7 x 7 area processing for both enhancement and texture development. The Institute has recently acquired a high precision hardcopy color device for improved output capability and has installed a real time TV solid state refresh monitor and display at ARPA headquarters. This allows recent pictorial results to be made available over the ARPANET. Any and all contractors can make use of this device with software devices available from the Institute. Finally this past six months have witnessed the graduation of one Ph.D. student and numerous Institute personnel publications. The Table of Contents of our up coming semi-annual report is listed below and provides insight into current projects. Interested readers are directed to that report (USCIPI No. 800).

TABLE OF CONTENTS

SYMBOLIC PROCESSING ALGORITHM RESEARCH COMPUTER
A Progress Report

Walling R. Cyre, Gale R. Allen, Pete G. Juetten

Control Data Corporation
Minneapolis, Minnesota

INTRODUCTION

This report summarizes the progress on a high-performance, microprogrammed computer called SPARC (Symbolic Processing Algorithm Research Computer), which was started by ARPA in 1977. Although a gap in funding occurred, this work has been continued by CDC in conjunction with Carnegie-Mellon University under an internal research and development program. ARPA funding has now been reestablished. The primary effort has been focused on the development of a set of design specifications from the system of desired architectural features reported earlier [1].

The machine organization of SPARC is based on the concept of a set of specialized Functional Units which communicate via a high-bandwidth, multiport switch. From the machine organization point of view, all Functional Units, including the Control Unit and I/O Units, are indistinguishable, except for the number of input and output ports which each presents to the Switch. The activities of the computer are governed by a Program Memory and a Connect Memory, both of which are located in the Control Unit. Two Program Memory instruction formats are defined. One instruction type has seven fields. The first field is used to issue activity or enable signals to the Functional Units, and the second field is a data or emit field. The next four fields may be used to select the operations to be performed by any four of the Functional Units, including the Control Unit. The final field specifies an address in the Connect Memory. The second instruction type is used to overlay the Connect Memory. The Connect Memory is used to store patterns of switch closures for interconnecting the Functional Units.

METHODOLOGY

The primary problems addressed in the effort reported here have been the specification of Functional Unit operation sets and the mapping of Functional Unit input ports onto Switch output ports. The tools which are being used to solve these problems include benchmarking and gate-level simulation. The approach has been to program a number of algorithms against the preliminary design (which included preliminary operation sets for each Functional Unit) and identify desirable changes in the design. In closely coupled efforts, the design feasibility and cost of each desired design modification was evaluated using gate-level simulation methods. Although these studies are incomplete at this time, significant results have been obtained, particularly in the area of Functional Unit operation sets.

RESULTS

First, the need for a general-register File Unit was identified. One of the original specifications on the SPARC was that it perform well at both the signal and symbol levels. In signal-level image processing tasks, the machine can be used to considerable advantage by cascading Functional Units to form pipelines. The File Unit is used to realize the small delays necessary in programming tight pipes.

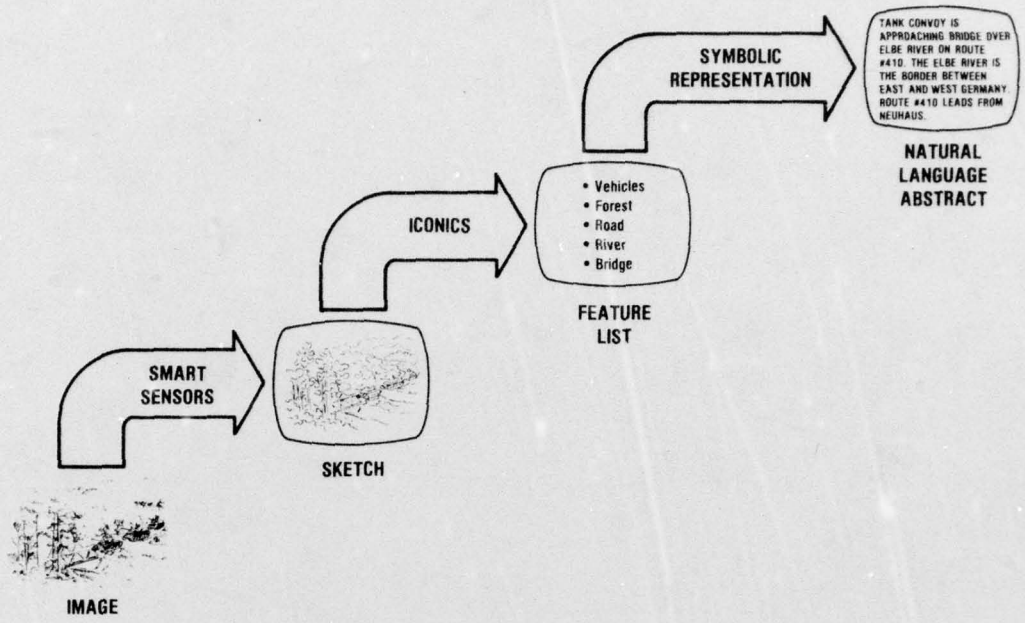
A second major result of the study was the integration of the Shift/Mask Unit and the Boolean Unit. This integration allows a higher utilization of the hardware, and was determined to be feasible through gate-level simulation. In addition to these modifications in the machine organizations, a number of improvements in the operation sets of the other Functional Units were made. These changes ranged from an additional multiplication mode in the Multiply Unit to a restructuring of the addressing mechanism of the Data Memory Units. Other modifications tending to improve performance were found, but were not adopted because they led to marginal timing situations or were not cost effective.

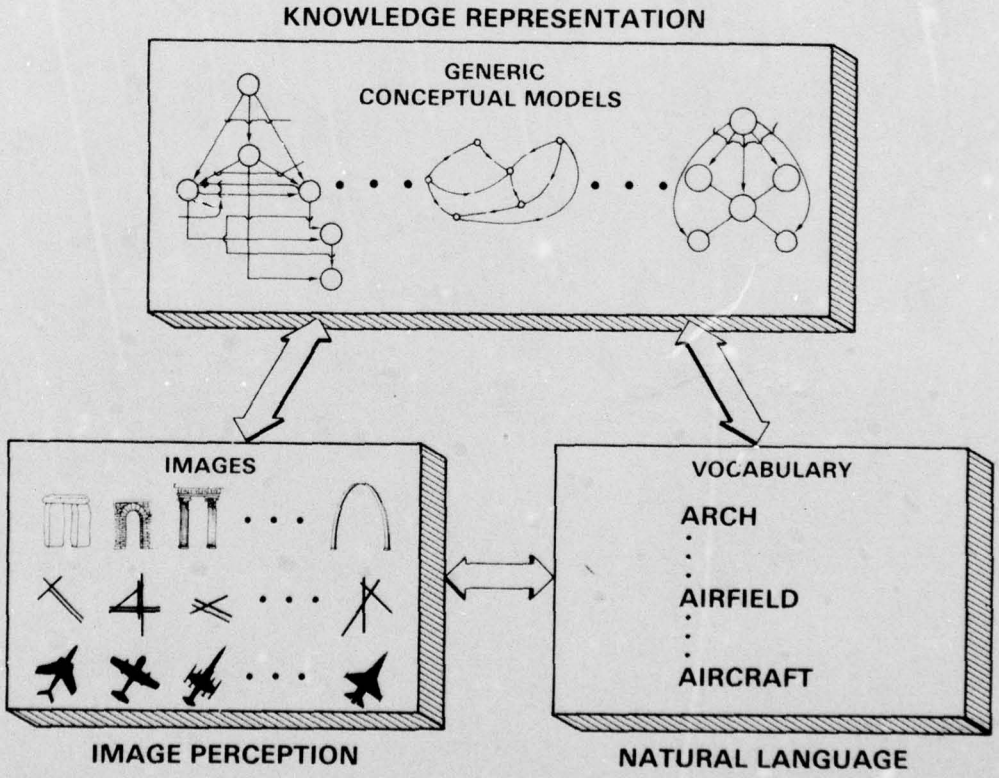
CONCLUSION

This effort is continuing with emphasis on optimizing the mapping from Functional Unit inputs to Switch outputs for conflict minimization, and on defining the mapping between detectable machine status signals and the conditions on which branching may be programmed. This method of benchmarking and simulation has been found to be a powerful tool for progressing from the preliminary architectural specifications to the hardware design specifications.

REFERENCES

[1] P. Juetten, G. Allen, R. Hon, and R. Reddy, "An Image Processor Architecture". Proc. Image Understanding Workshop, Palo Alto, CA October 20-21, pg. 12-18.


SYMBOLIC REPRESENTATION
TANK CONVOY IS APPROACHING BRIDGE OVER ELBE RIVER ON ROUTE #410. THE ELBE RIVER IS THE BORDER BETWEEN EAST AND WEST GERMANY. ROUTE #410 LEADS FROM NEUHAUS.
NATURAL LANGUAGE ABSTRACT
ICONICS
• Vehicles
• Forest
• Road
• River
• Bridge
FEATURE LIST
SMART SENSORS
SKETCH
IMAGE


KNOWLEDGE REPRESENTATION
GENERIC CONCEPTUAL MODELS
IMAGES
VOCABULARY
ARCH
AIRFIELD
AIRCRAFT
IMAGE PERCEPTION
NATURAL LANGUAGE